Recent Developments in Statistical Dialogue Systems

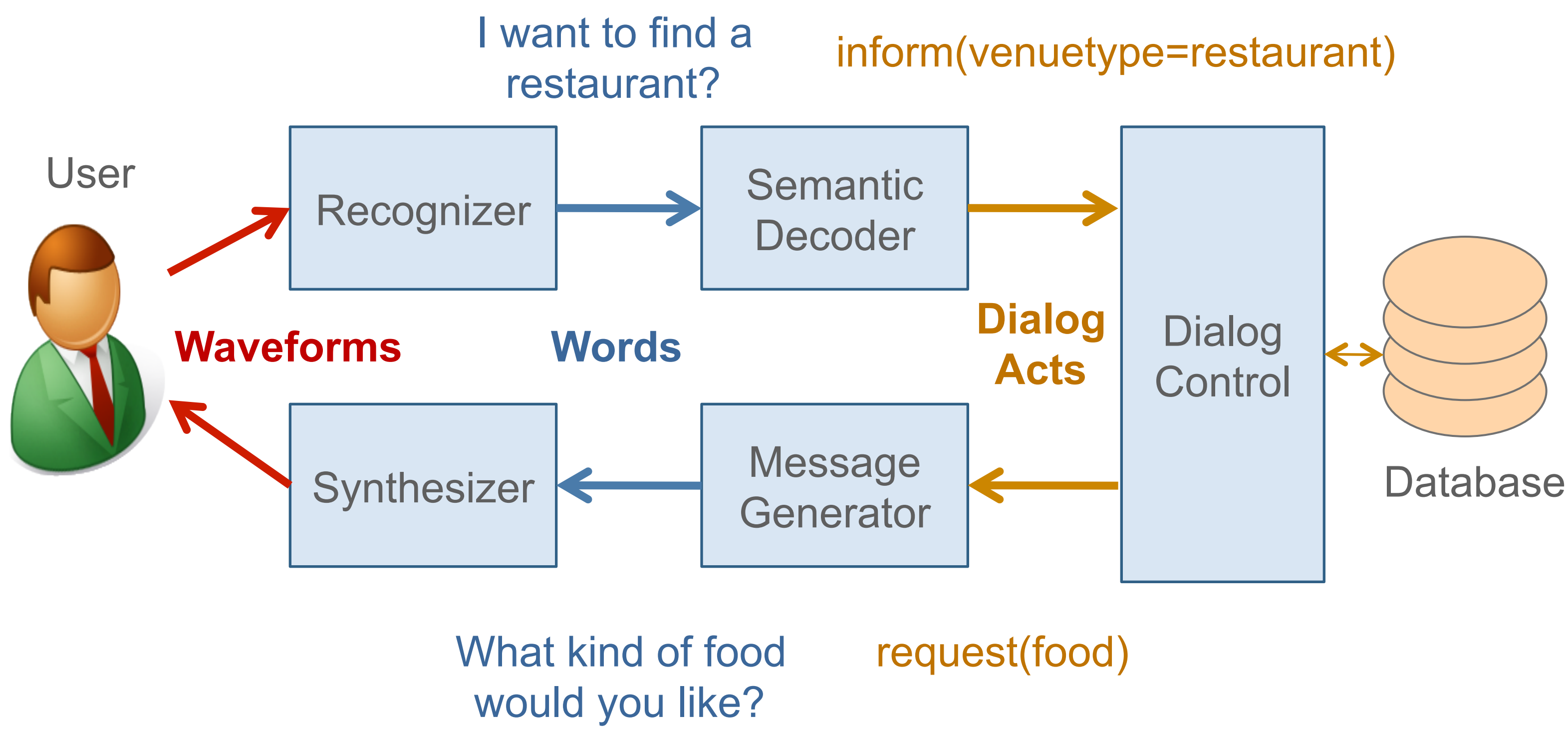
Steve Young

PARLANCE

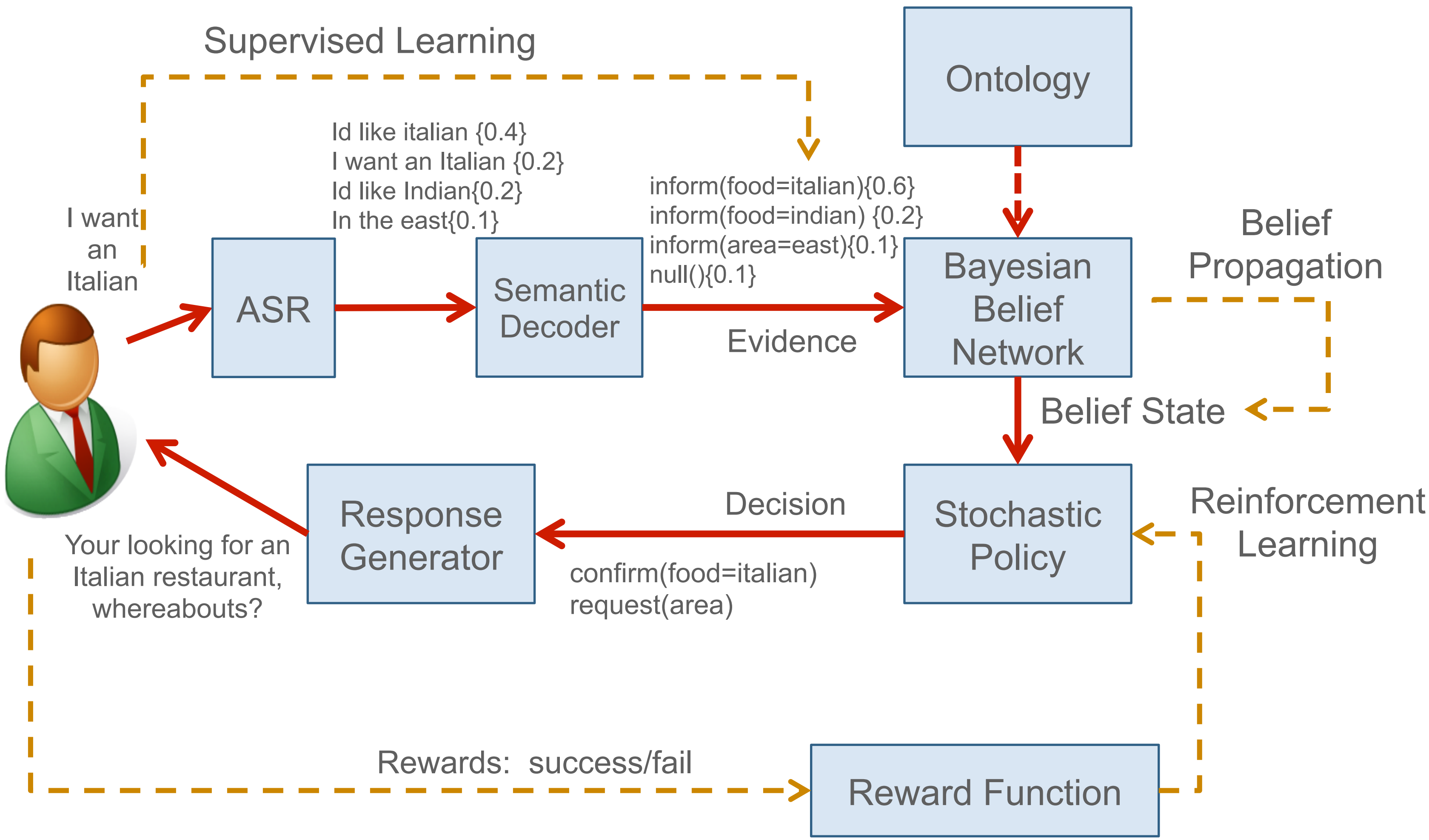
Machine Intelligence Laboratory
Information Engineering Division
Cambridge University Engineering Department
Cambridge, UK

Contents

Spoken Dialog Systems (SDS)



A Statistical Spoken Dialogue System



Partially Observable Markov Decision Process (POMDP)

The POMDP SDS Framework

observation

semantic decoder

speech recogniser

$$o_t = p(u_t \mid x_t) = \sum_{w_t} p(u_t \mid w_t)\, p(w_t \mid x_t)$$

user acts, speech, user acts, words, words, speech

belief state

$$b_t(s_t) = \eta\, p(o_t \mid s_t, a_{t-1}) \sum_{s_{t-1}} P(s_t \mid s_{t-1}, a_{t-1})\, b_{t-1}(s_{t-1})$$

state, state, action

system action

policy parameters, belief state features

$$a_t \sim \pi(b_t, a_t) = \frac{e^{\theta \cdot \phi_{a_t}(b_t)}}{\sum_a e^{\theta \cdot \phi_a(b_t)}}$$

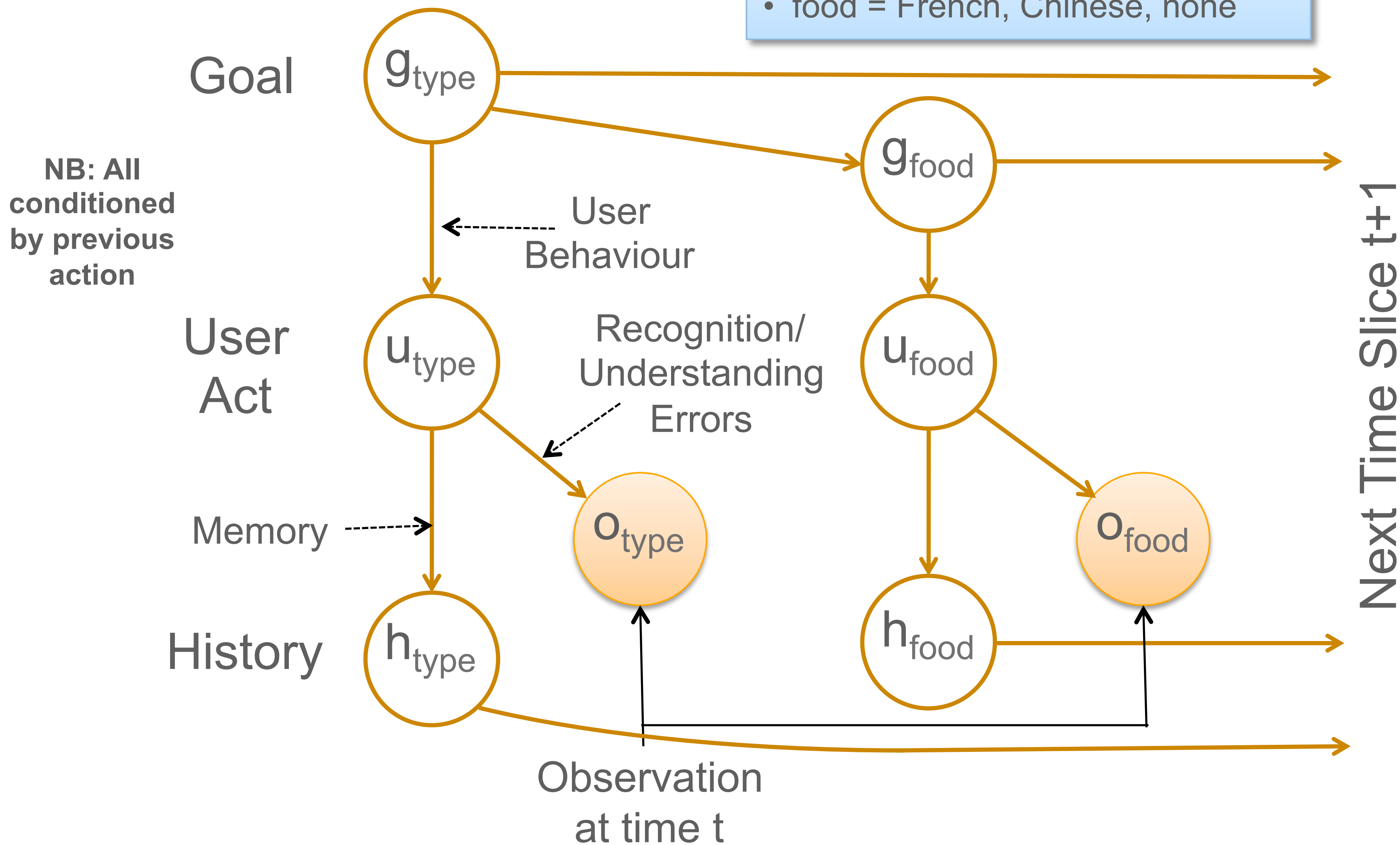
reward function

$$R = E\left\{ \sum_{s_t} r(s_t, a_t)\, b_t(s_t) \right\}$$

Maximise θ

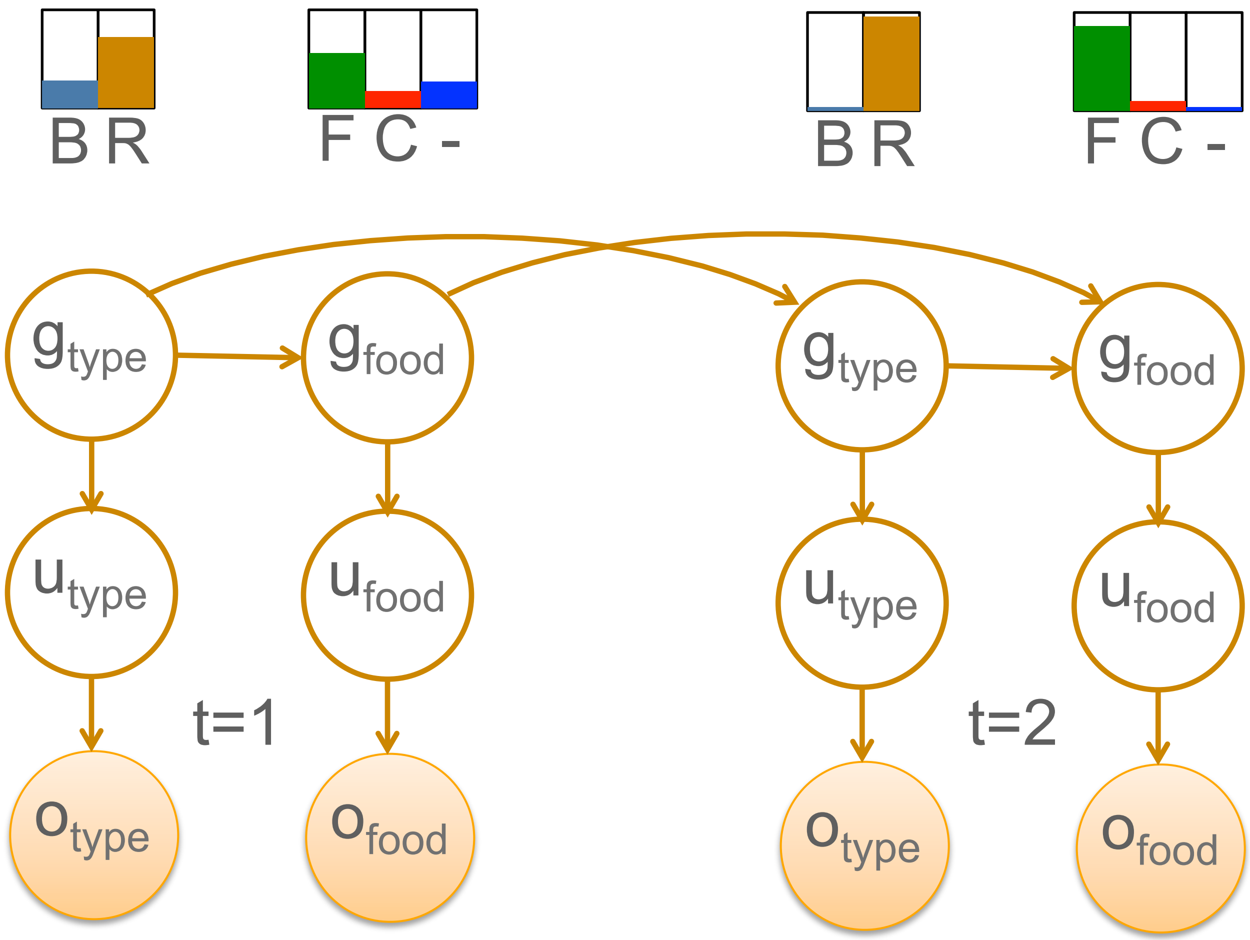
Dialogue State

Tourist Information Domain
- type = bar,restaurant
- food = French, Chinese, none



J. Williams (2007). "POMDPs for Spoken Dialog Systems." Computer Speech and Language 21(2)

Belief Monitoring (Tracking)

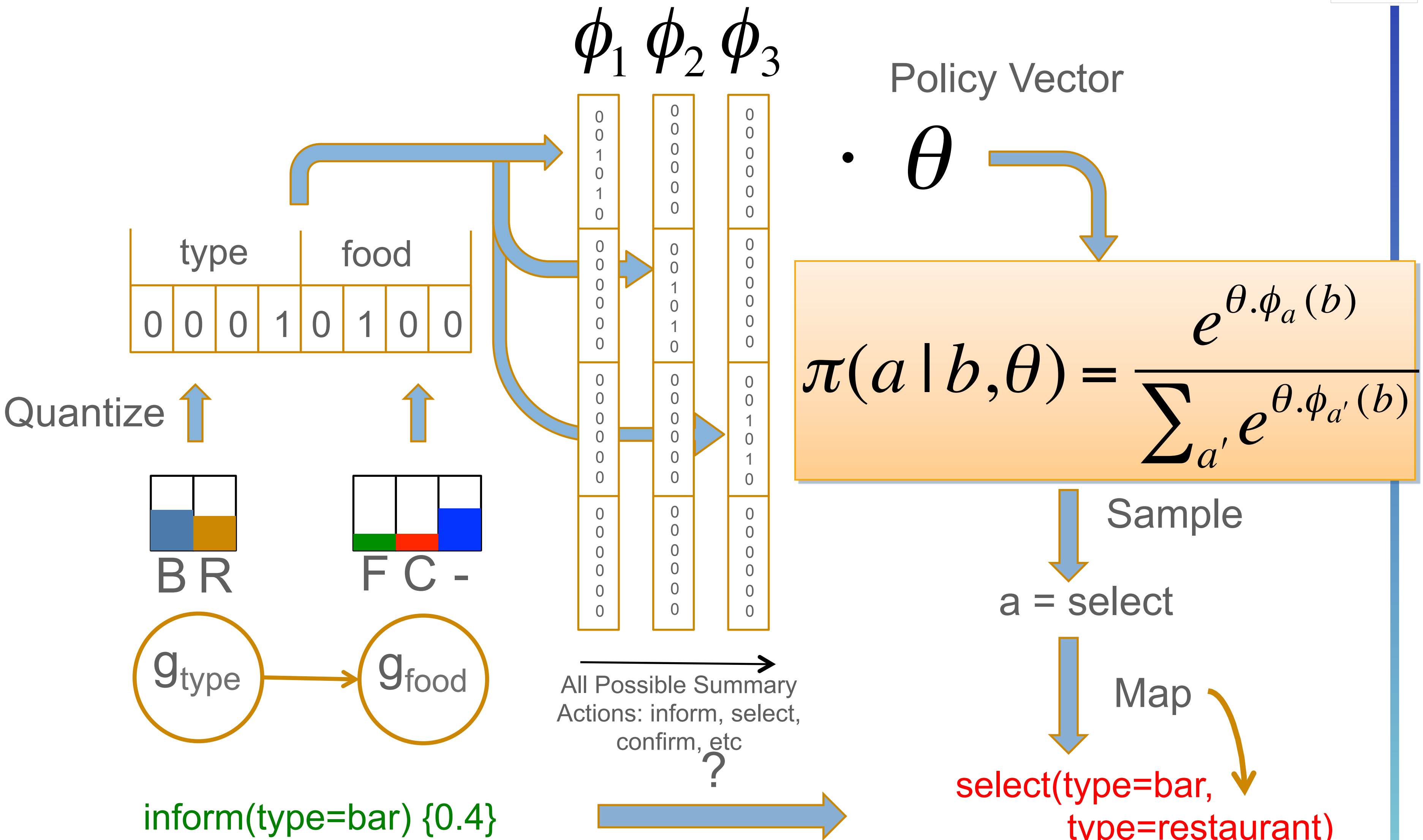


inform(type=bar,
food=french) {0.6}
inform(type=restaurant,
food=french) {0.3}

confirm(type=restaurant,
food=french)

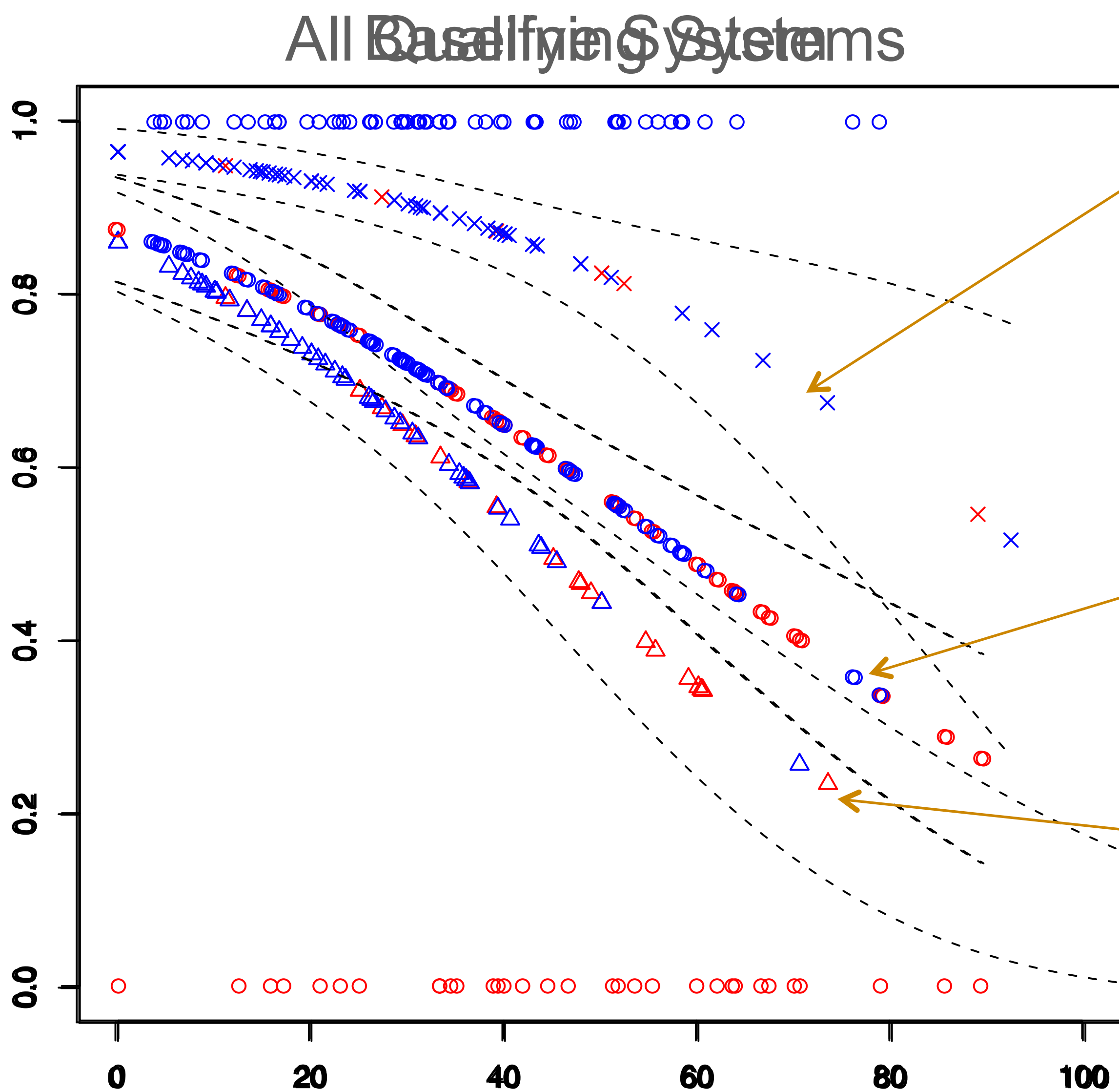
affirm() {0.9}

Choosing the next action – the Policy



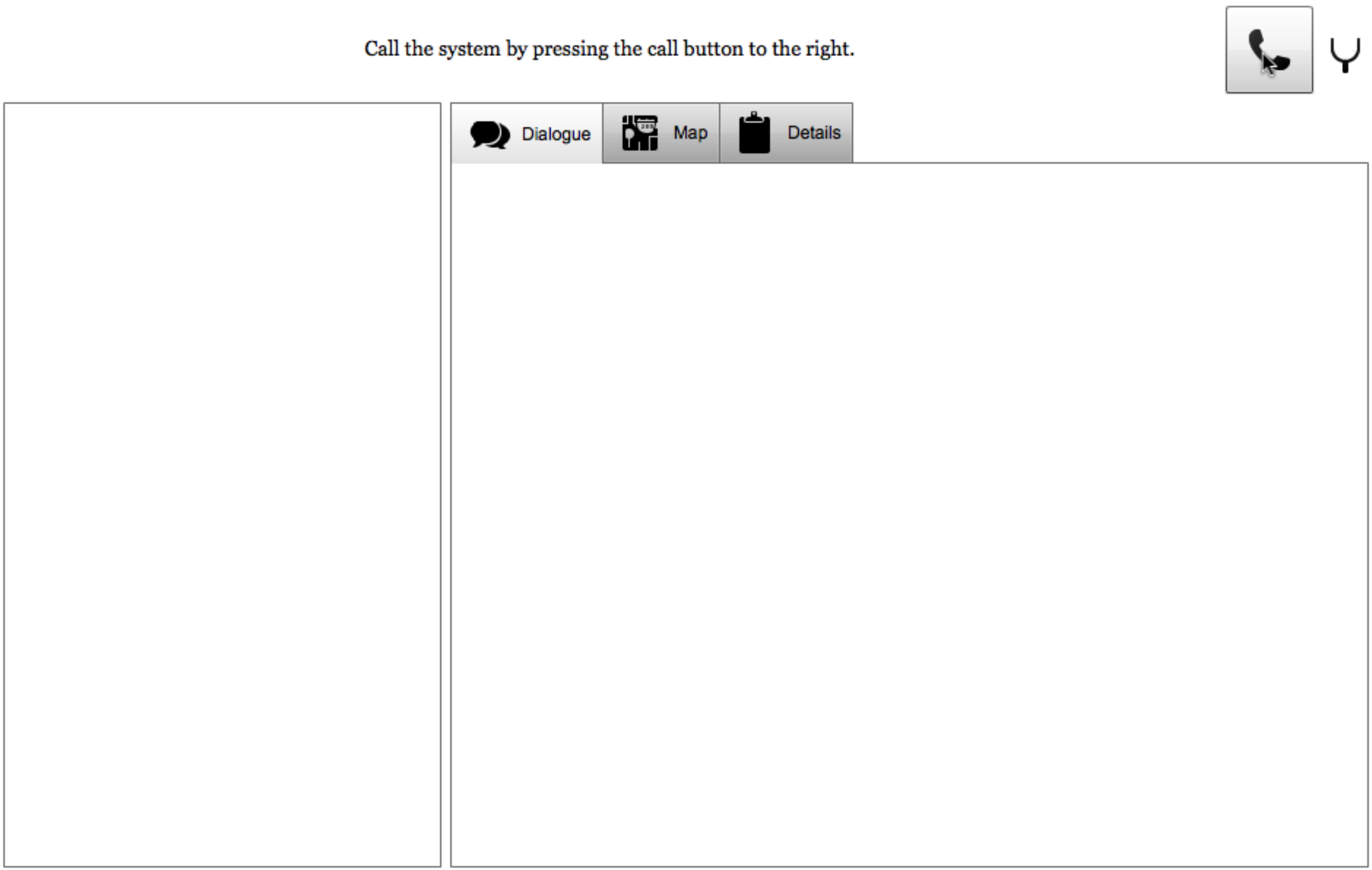
$$\pi(a \mid b,\theta) = \frac{e^{\theta.\phi_a(b)}}{\sum_{a'} e^{\theta.\phi_{a'}(b)}}$$

Let's Go 2010 Control Test Results



B. Thomson "Bayesian Update of State for the Let's Go Spoken Dialogue Challenge." SLT 2010.

Demo of Cambridge Restaurant Information

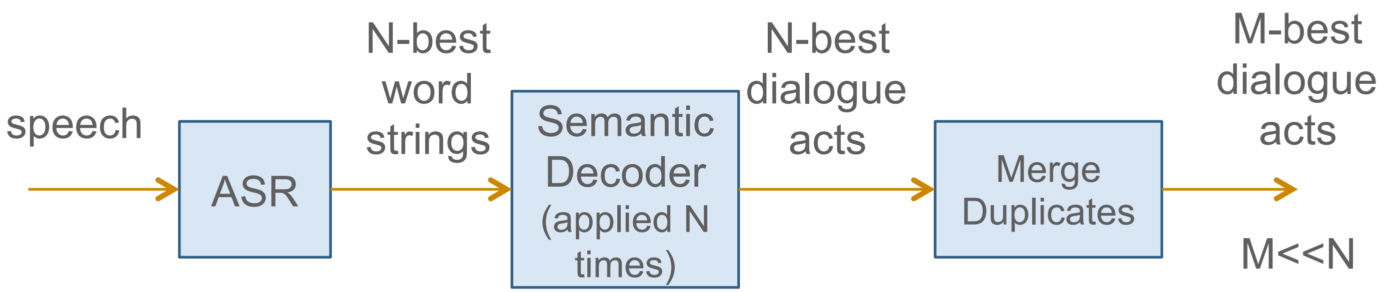


Issues with the 2010 System Design

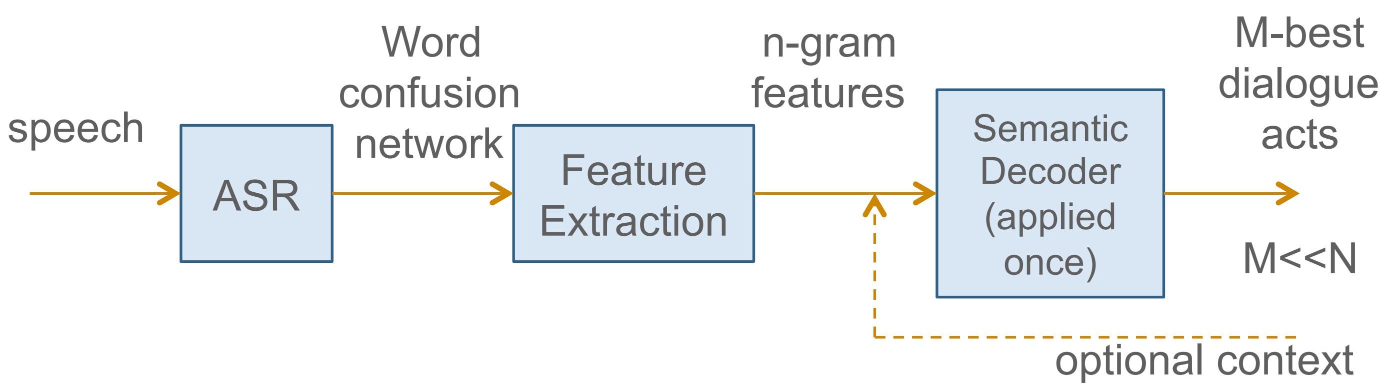
- Poor coverage of N-best of semantic hypotheses
- Hand-crafting of summary belief space
- Slow policy optimisation and reliance on user simulation
- Dependence on hand-crafted dialogue model parameters
- Dependence on static ontology/database

N-best Semantic Decoding

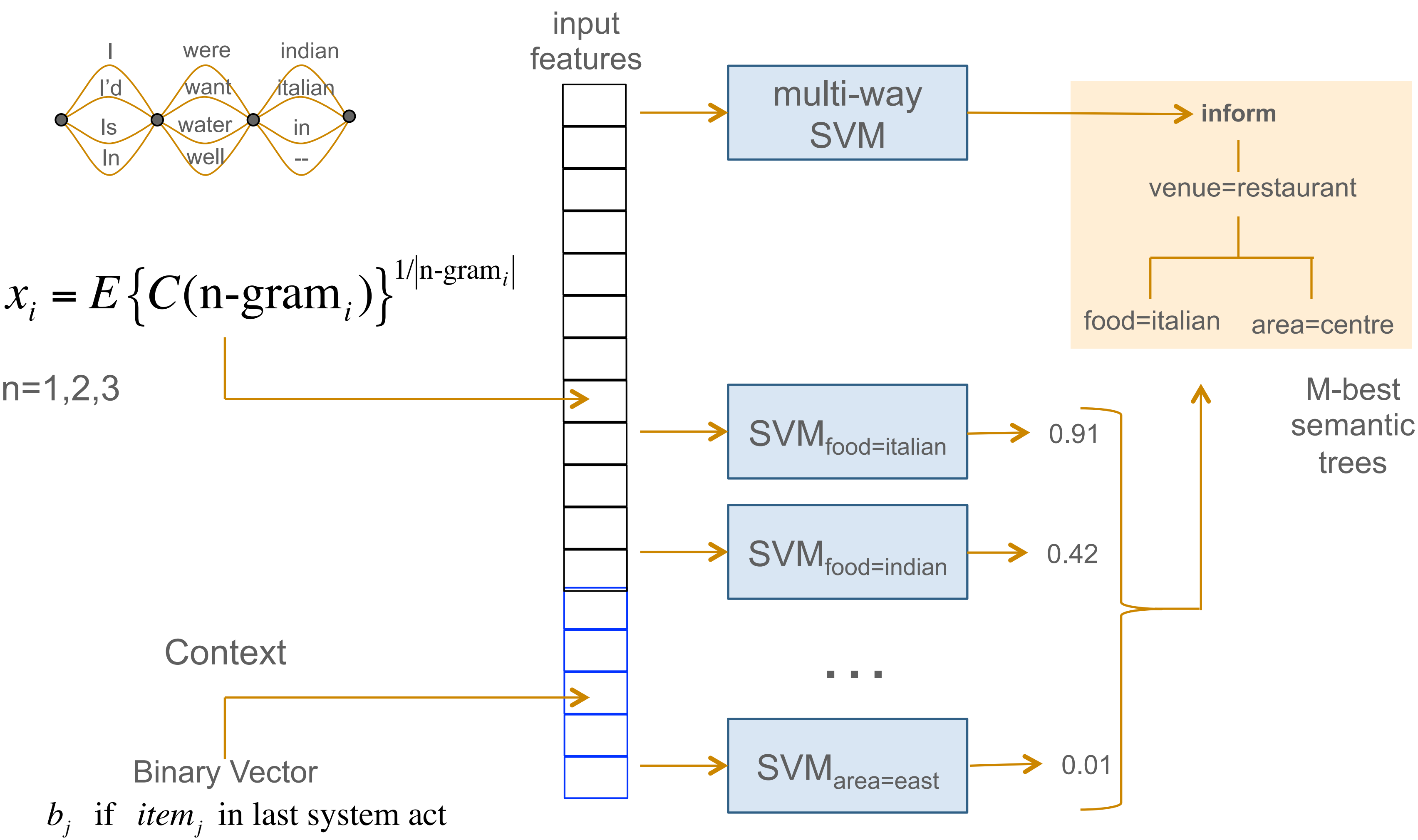
Conventional N-best decoding



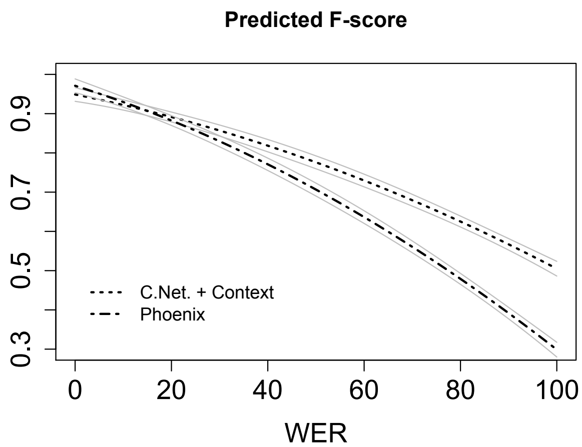
Confusion Network decoding

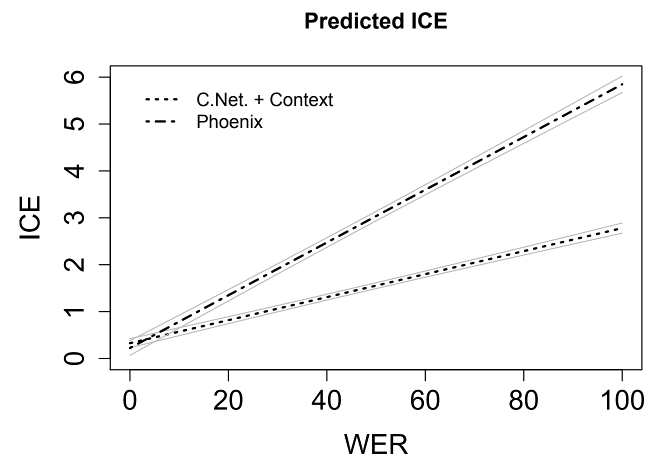


Confusion Network Decoder (Mairesse/Henderson)



Confusion Network Decoder Evaluation





Comparison of item retrieval on corpus of 4.8k utterances

N-best hand-crafted Phoenix decoder vs Confusion network decoder (trained on 10k utterances)

Live Dialogue System

	N-best Phoenix	Confusion Net
F-score	0.80	0.82
ICE	2.02	1.26
Average Reward	10.6	11.15

"Discriminative Spoken Language Understanding using Word Confusion Networks", Henderson et al, IEEE SLT 2012

Policy Optimization

Policy parameters chosen to maximize expected reward

$$J(\theta) = E\left[\frac{1}{T}\sum_t r(s_t, a_t) \mid \pi_\theta\right]$$

Natural gradient ascent works well

$$\tilde{\nabla} J(\theta) = F_\theta^{-1} \nabla J(\theta)$$

Fisher Information Matrix

Gradient is estimated by sampling dialogues and in practice Fisher Information Matrix does not need to be explicitly computed.

This is the Natural Actor Critic Algorithm.

However,

A) slow (~100k dialogues) and
B) requires summary space approximations

J. Peters and S. Schaal (2008). "Natural Actor-Critic." Neurocomputing 71(7-9)

Q-functions and the SARSA algorithm

Traditional reinforcement learning is commonly based on finding the optimal Q function:

$$Q^*(b,a) = \max_{\pi}\left[E_{\pi}\left\{\sum_{\tau=t+1}^{T} r(b_\tau, a_\tau)\right\}\right]$$

The optimal deterministic policy is then simply

$$\pi^*(b) = \underset{a}{\operatorname{argmax}}\left[Q^*(b,a)\right]$$

Q^* can be found sequentially using the SARSA algorithm

b = b_0; choose action a e-greedily from π(b)
For each dialogue turn
    Take action a, observe reward r and next state b'
    choose action a' e-greedily from π(b')
    Q(b, a) = Q(b, a) +λ[Q(b', a') – (Q(b, a) – r)]
    b = b'; a = a'
end

Eventually, $Q \rightarrow Q^*$

Gaussian Process based Learning – Milica Gasic

For POMDPs, the belief space is continuous and direct representations of Q are intractable. However, Q can be approximated as a zero mean Gaussian process by designing a *kernel* to represent the correlations between points in belief x action space. Thus:

$$Q(b,a) \sim GP\big(0, k\big((b,a),(b,a)\big)\big)$$

Given a sequence of state-action pairs

$$B_t = \left[(b_0,a_0),(b_1,a_1),..,(b_t,a_t)\right]' \quad \text{and rewards} \quad r_t = \left[r_0,r_1,..,r_t\right]'$$

there is a closed form solution for the posterior:

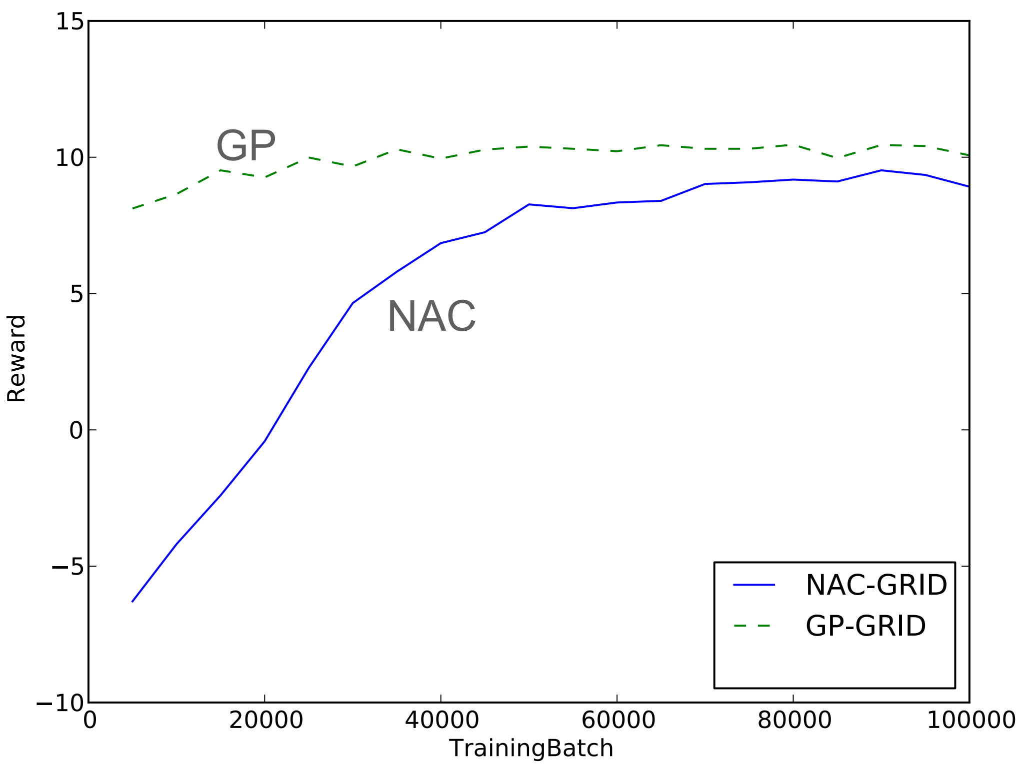
$$Q(b,a) \mid B_t, r_t \sim N\big(\bar{Q}(b,a), \mathrm{cov}\big((b,a),(b,a)\big)\big)$$

This suggests a SARSA-like sequential optimisation:

```
b = b0; choose action a e-greedily from Q̄(b,a)
For each dialogue turn
        Take action a, observe reward r and next state b'
        choose action a' e-greedily from Q̄(b',a')
        Update the posterior covariance estimate
        b = b'; a = a'
end
```

Benefits of GP-SARSA

- sequential estimation of distribution of Q (not Q itself)
- each new data point can impact on whole distribution via the covariance function
  → very efficient use of training data
- much faster learning than gradient methods such a Natural Actor Critic (NAC)



TownInfo
System

ε-greedy exploration

$$a = \begin{cases} \underset{a}{\operatorname{argmax}}\left[\bar{Q}(b,a)\right] & \text{with prob } 1-\varepsilon \\ \text{random action} & \text{with prob } \varepsilon \end{cases}$$

Benefits of GP-SARSA

- variance of Q is known at each stage → more intelligent exploration:

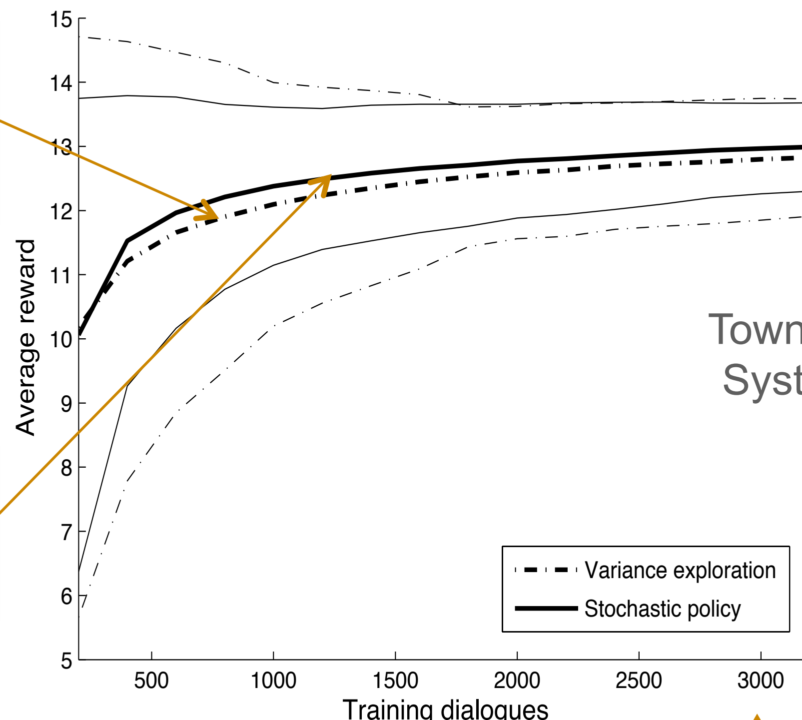
Variance exploration

$$a = \begin{cases} \arg\max_a \left[\bar{Q}(b,a)\right] & \text{with prob } 1-\varepsilon \\ \arg\max_a \left[\mathrm{cov}((b,a),(b,a))\right] & \text{with prob } \varepsilon \end{cases}$$

Stochastic policy

$$Q^i(b,a_i) \sim N\left(\bar{Q}(b,a_i), \mathrm{cov}((b,a_i),(b,a_i))\right)$$

$$a = \arg\max_{a_i}\left[Q^i(b,a_i)\right]$$

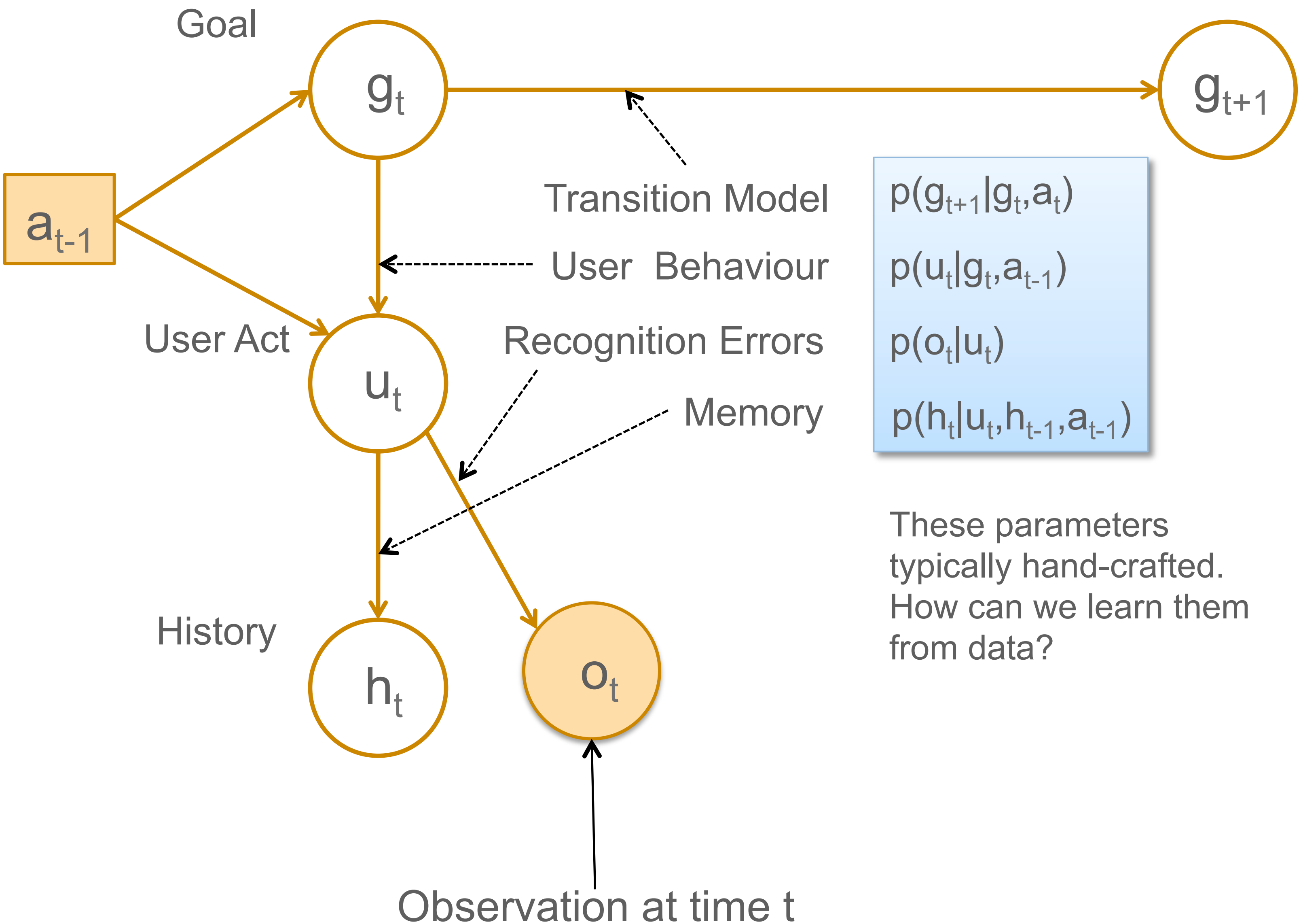


TownInfo System

Well trained within 3k dialogues
And summary space mapping no longer needed

“On-line policy optimisation of SDS via live interaction with human subjects”, Gasic et al, ASRU 2011
“Gaussian processes for policy optimisation of large scale POMDP-based SDS”, Gasic et al, SLT 2012.

Parameter Estimation – Blaise Thomson



$p(g_{t+1}|g_t,a_t)$

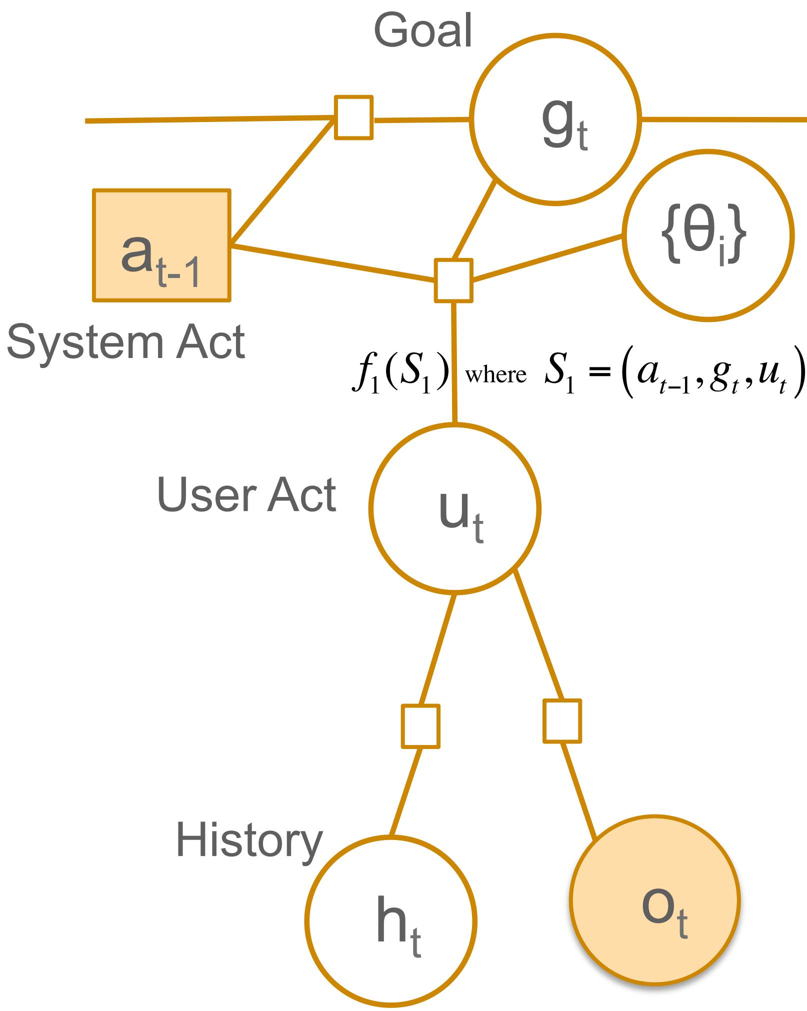
$p(u_t|g_t,a_{t-1})$

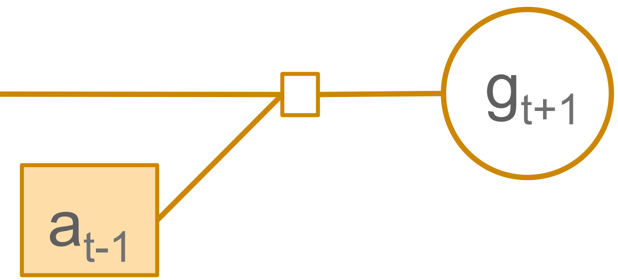
$p(o_t|u_t)$

$p(h_t|u_t,h_{t-1},a_{t-1})$

These parameters typically hand-crafted. How can we learn them from data?

Factor Graphs and Expectation Propagation



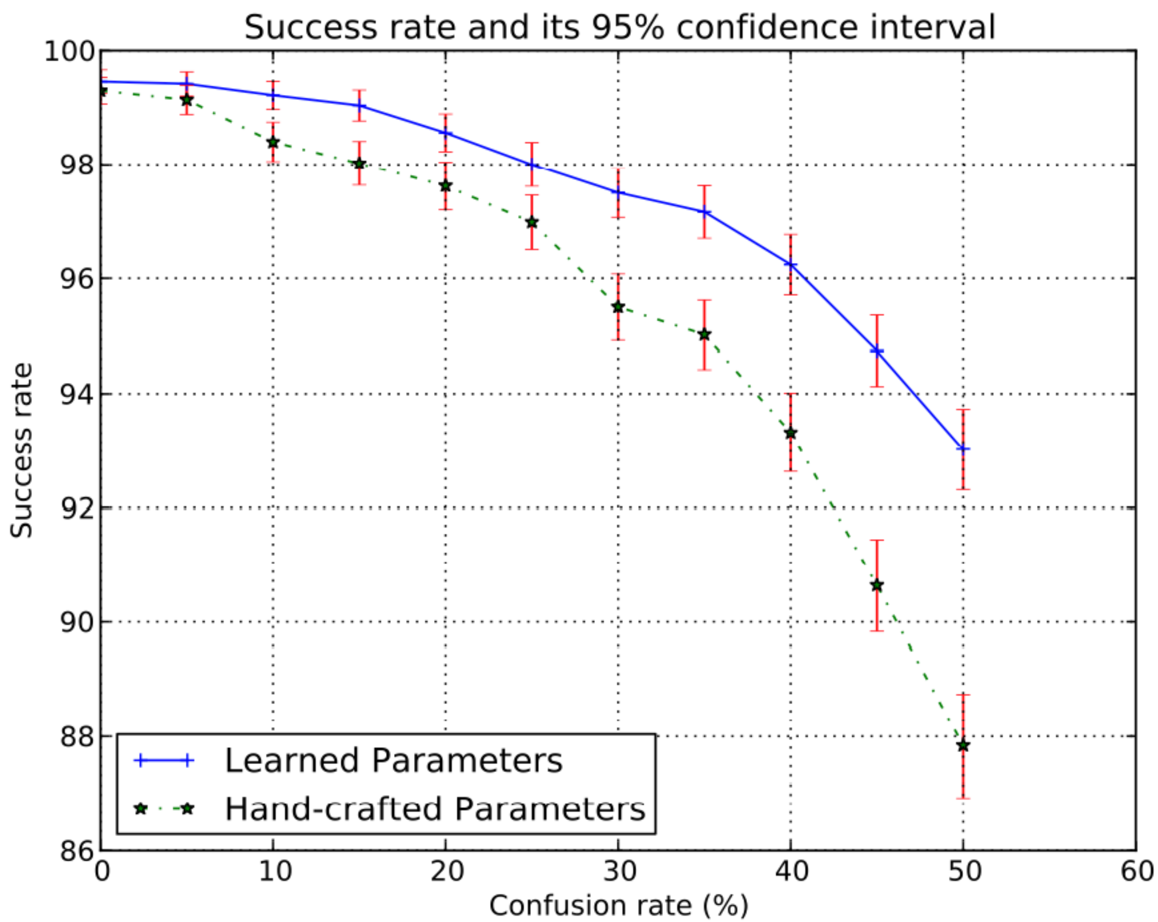


Posterior

$$b(s_t \mid o_t, s_{t-1}, a_{t-1}) \propto \prod_{k=1}^{K} f_k(S_k)$$

- exact computation is intractable
- can be approximated using belief propagation
- we use Expectation Propagation (EP)
- using EP factors can be discrete & continuous
- *hence, parameters can be added and updated simultaneously*

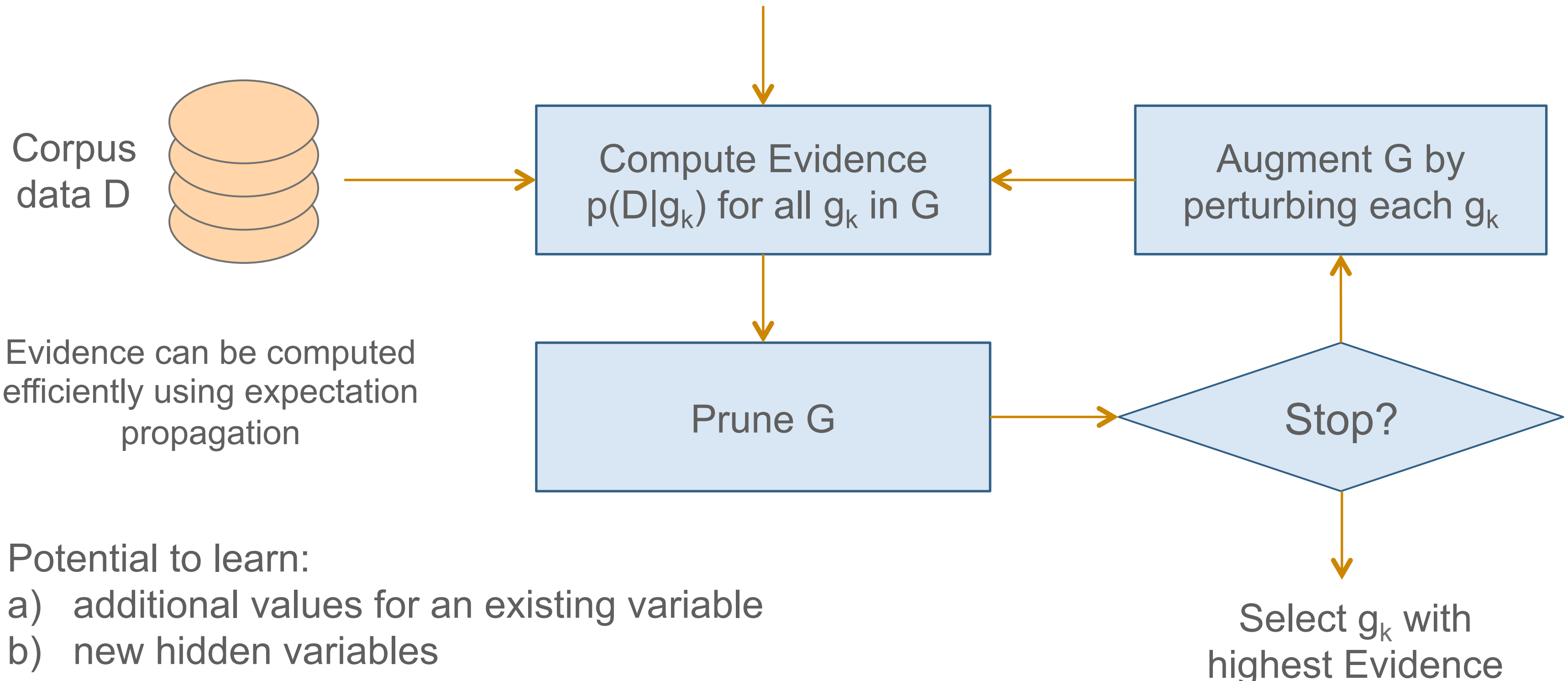
Effect of Parameter Learning on TownInfo System



Structure Learning

The ability to learn parameters can be extended to learn structure.

Let $G = \{ g_k \}$ be a set of Bayesian Networks (or Factor Graphs):



Evidence can be computed efficiently using expectation propagation

Potential to learn:

a) additional values for an existing variable
b) new hidden variables
c) new links between variables

PARLANCE

Conclusions

- Statistical Dialogue Systems based on POMDPs are viable, offer increased robustness to noise and require no hand-crafting
- Good progress is being made on increasing accuracy and speeding up learning
- Learning directly on human users rather than depending on user simulators is now possible
- Current systems are built from static ontologies for closed domains
- Next steps will include building more flexible systems capable of dynamically adapting to new information content.

Acknowledgement - this work is the result of a team effort:
Catherin Breslin, Milica Gasic, Matt Henderson, Dongho Kim,
Martin Szummer, Blaise Thomson, Pirros Tsiakoulis
and former members of the CUED Dialogue Systems Group