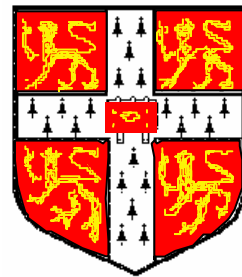


# Dialogue Management using Partially Observable Markov Decision Processes



Jason D. Williams

Machine Intelligence Laboratory  
Cambridge University Engineering Department

# Outline



- Model: Proposed POMDP representation of dialogue
- Part 1: Comparison with baselines
  - Example: 3-place problem
- Part 2: Scaling up with the “summary POMDP” method
  - Example: 1000-place problem
- Conclusions & future work

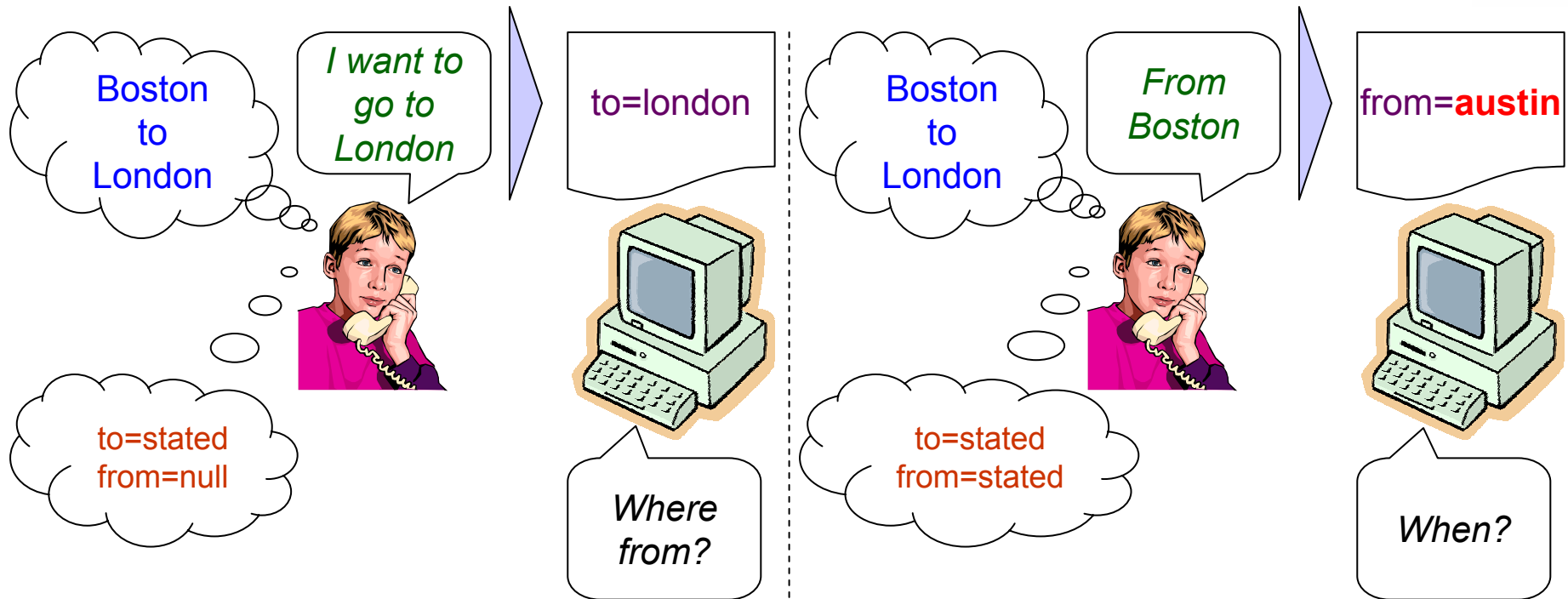
**Material in this talk is joint work with Pascal Poupart and Steve Young**

Jason D. Williams, Pascal Poupart, and Steve Young (2005). *Factored Partially Observable Markov Decision Processes for Dialogue Management*. 4th Workshop on Knowledge and Reasoning in Practical Dialog Systems, International Joint Conference on Artificial Intelligence (IJCAI), August 2005, Edinburgh.

Jason D. Williams, Pascal Poupart, and Steve Young (2005). *Partially Observable Markov Decision Processes with Continuous Observations for Dialogue Management*. In Proceedings of the 6th SigDial Workshop on Discourse and Dialogue, September 2005, Lisbon.



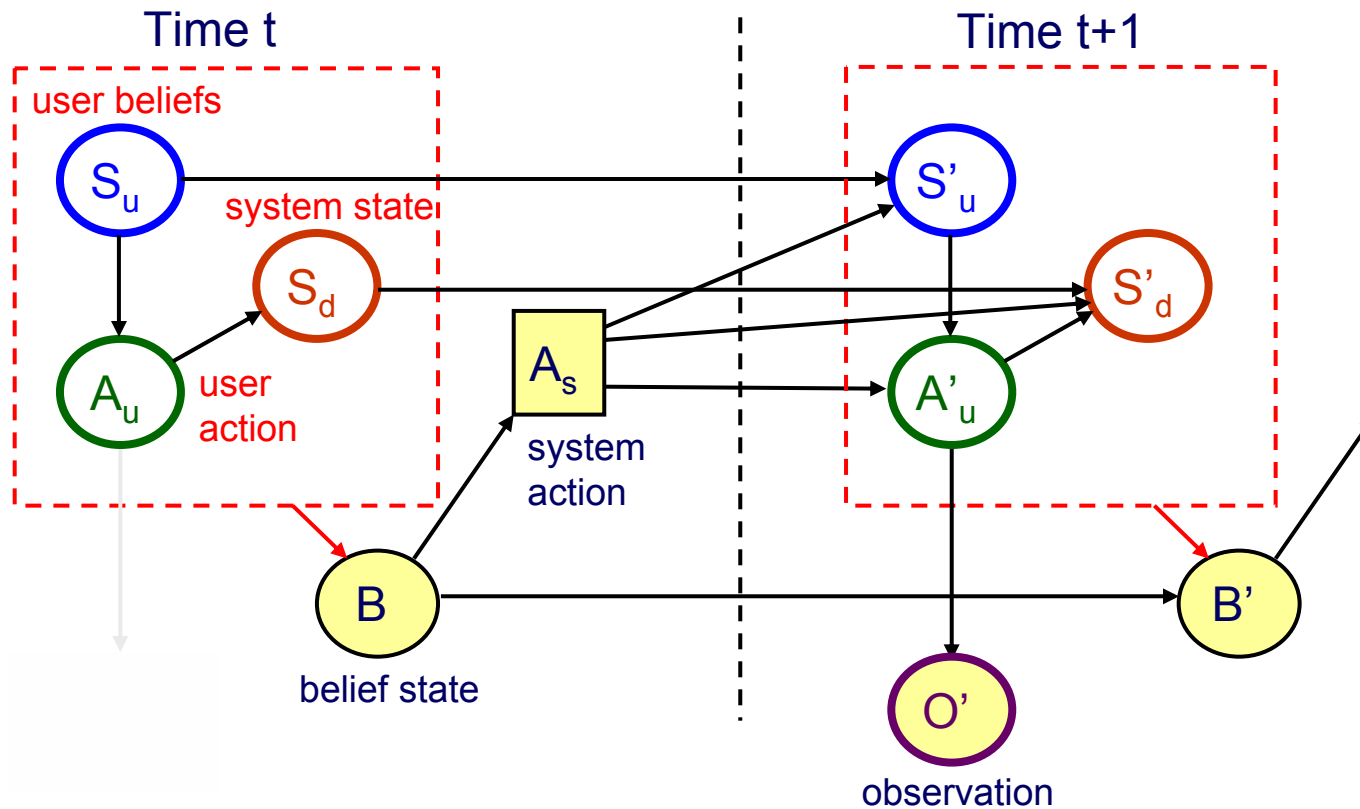
# Idealized Human-Computer Dialog



- Current state = dialogue modelling
- System action selection = dialogue management
- Dialogue state is *unobserved*:
  - User's goal
  - User's (real) action
  - Conversation state
- Inferences via *observation*:
  - May contain **errors**



# Decompose state variable into 3 “models”



# What's really different?

Typical approaches encode uncertainty through explicit state variables. Eg.

|                  |           |
|------------------|-----------|
| Num Pizza's:     |           |
| Variable Status: | undefined |
| Confidence:      |           |

Sys: How many pizza's?  
User: **Three, please.**

$$A_s = \Pi(S)$$

|                  |         |
|------------------|---------|
| Num Pizza's:     | 2       |
| Variable Status: | defined |
| Confidence:      | medium  |

Sys: You want two pizza's?  
User: **No.**

$$A_s = \Pi(S)$$

|                  |           |
|------------------|-----------|
| Num Pizza's:     |           |
| Variable Status: | undefined |
| Confidence:      |           |

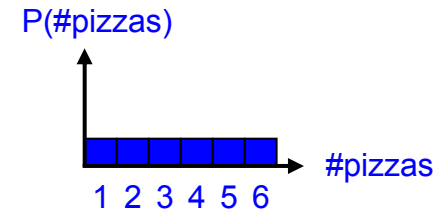
Sys: How many pizza's?  
User: **I want three.**

$$A_s = \Pi(S)$$

|                  |         |
|------------------|---------|
| Num Pizza's:     | 3       |
| Variable Status: | defined |
| Confidence:      | medium  |

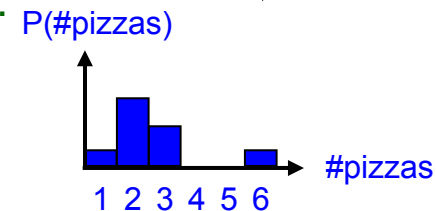
Sys: You want three pizza's?  
User: **Yes.**

Instead we can maintain a distribution over states – i.e., maintain *all possible states* – and update the distribution at each timestep



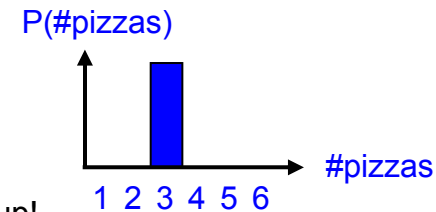
Sys: How many pizza's?  
User: **Three, please.**

$$A_s = \Pi(P(S))$$



Sys: Was that two or three?  
User: **Three.**

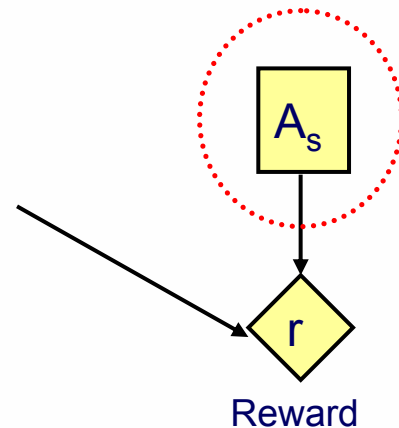
$$A_s = \Pi(P(S))$$



Sys: Coming right up!

**We can think of a belief state as a “cumulative” confidence measure over all user goals.**

# Toward dialogue management: control

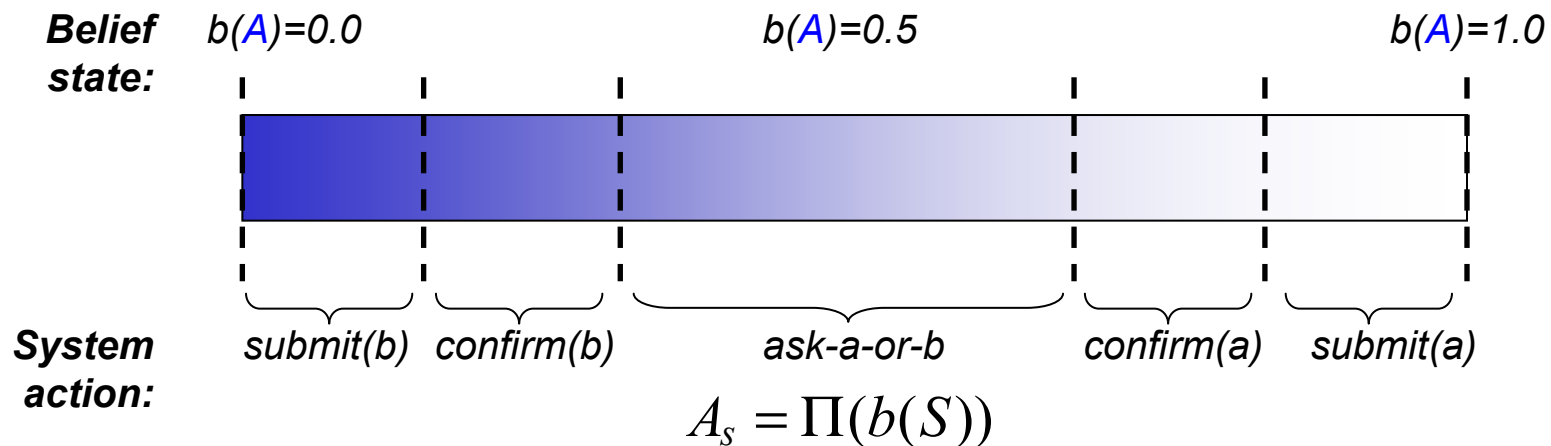


- Problem: how to choose  $a_s$ ?
  - Treat as a design problem (for human designers)
  - Specify  $r$ ; choose  $a_s$  to max. immediate reward (no planning)
  - Specify  $r$ ; choose  $a_s$  to max. cumulative reward (planning - POMDP)



# A policy as a partitioning

- A policy is a mapping: *situation*  $\rightarrow$  *action*
- One representation is a *partitioning* of belief space
- 2-dimensional example:
  - Simple state space with 2 user goals: **A & B**
  - Belief space can be written as  $b(A)$  (b/c  $b(B) = 1 - b(A)$ )
  - Policy shows action to take for each point in belief space



**This partitioning is produced by a POMDP optimization method**

# Outline



- Motivation
- Model: Proposed POMDP representation of dialogue
- Part 1: Comparison with baselines
  - Example: 3-place problem
- Part 2: Scaling up with the “summary POMDP” method
  - Example: 1000-place problem
- Conclusions & future work



## 3-place problem

User is travelling from  $x$  to  $y$  in a world with 3 cities,  $a$ ,  $b$ , and  $c$ .

- $s_u$  : The user's desired *from* and *to* cities – e.g.,  $(a,b)$
- $a_u, o$  : *yes, no, null, a, from-a, to-a, from-a-to-b*, etc.
- $s_d$  : Each slot can be *not stated, unconfirmed, or confirmed*
- $a_s$  : *greet, ask-from, ask-to, conf-from-a, conf-to-b, submit-a-b, fail*
  
- **User goal model**: User has a fixed goal throughout the dialogue
- **User action model**: User responds with varied but cooperative actions. Sometimes the user doesn't respond (*null*)
- **Conversation model**: Deterministically “promotes” slots
- **Speech recognition model** : Uniform errors with probability  $p_{err}$
- **Reward fn** : +10 correct submit, -10 wrong submit, -5 fail, -3 for confirming something which is *not stated*, -1 per-turn otherwise
  
- **Solve with *Perseus* (PBVI)**



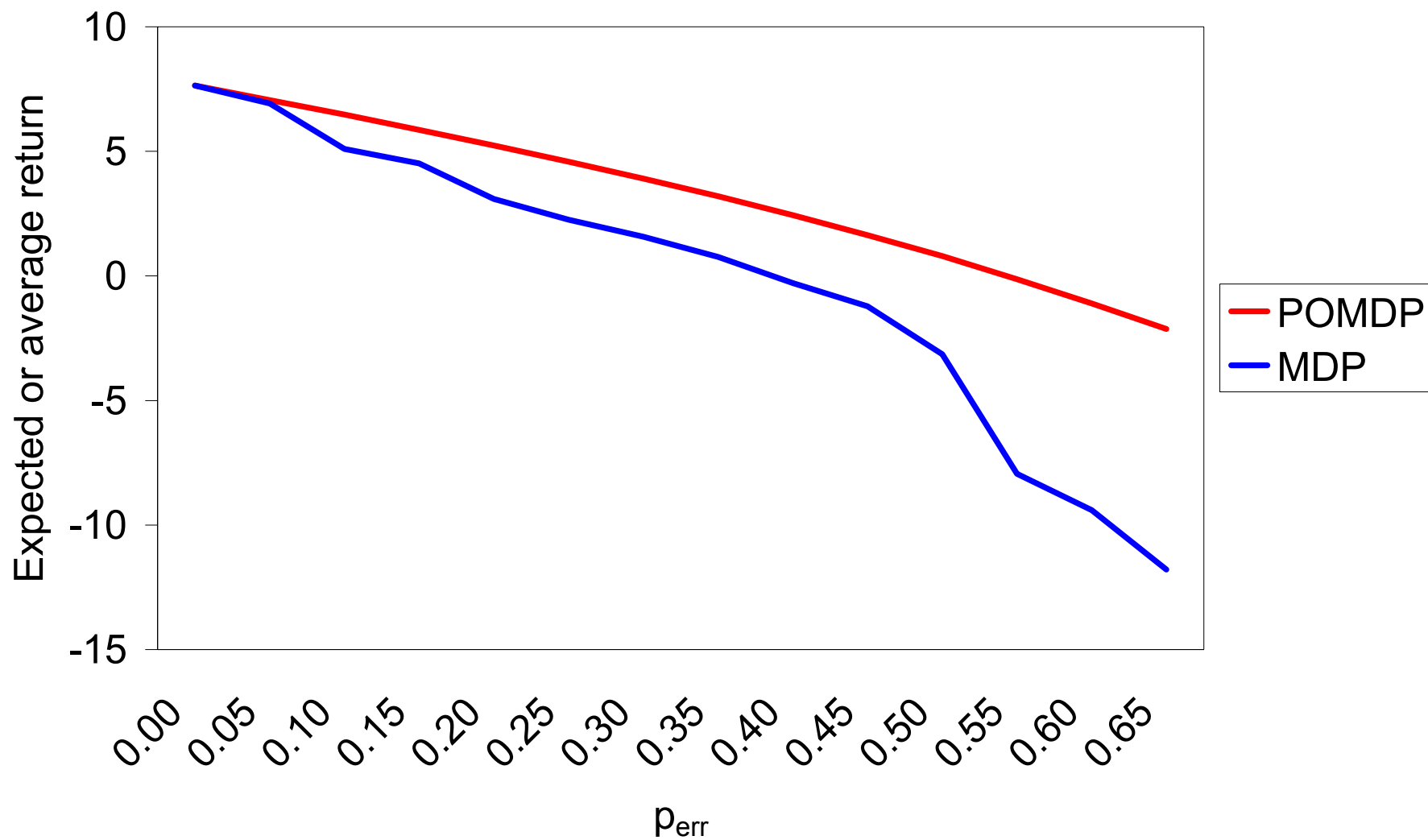
# Baseline 1: MDP

- Same system dynamics as in POMDP
- Pass observations to an MDP
- *From* and *to* slots can be “unknown”, “observed”, or “confirmed”
- State estimator using typical heuristics
  - Set a slot to “unknown” when user answers “no” to an explicit confirmation

Optimize using a standard technique (*Q-learning*)



# Results: POMDP vs. MDP

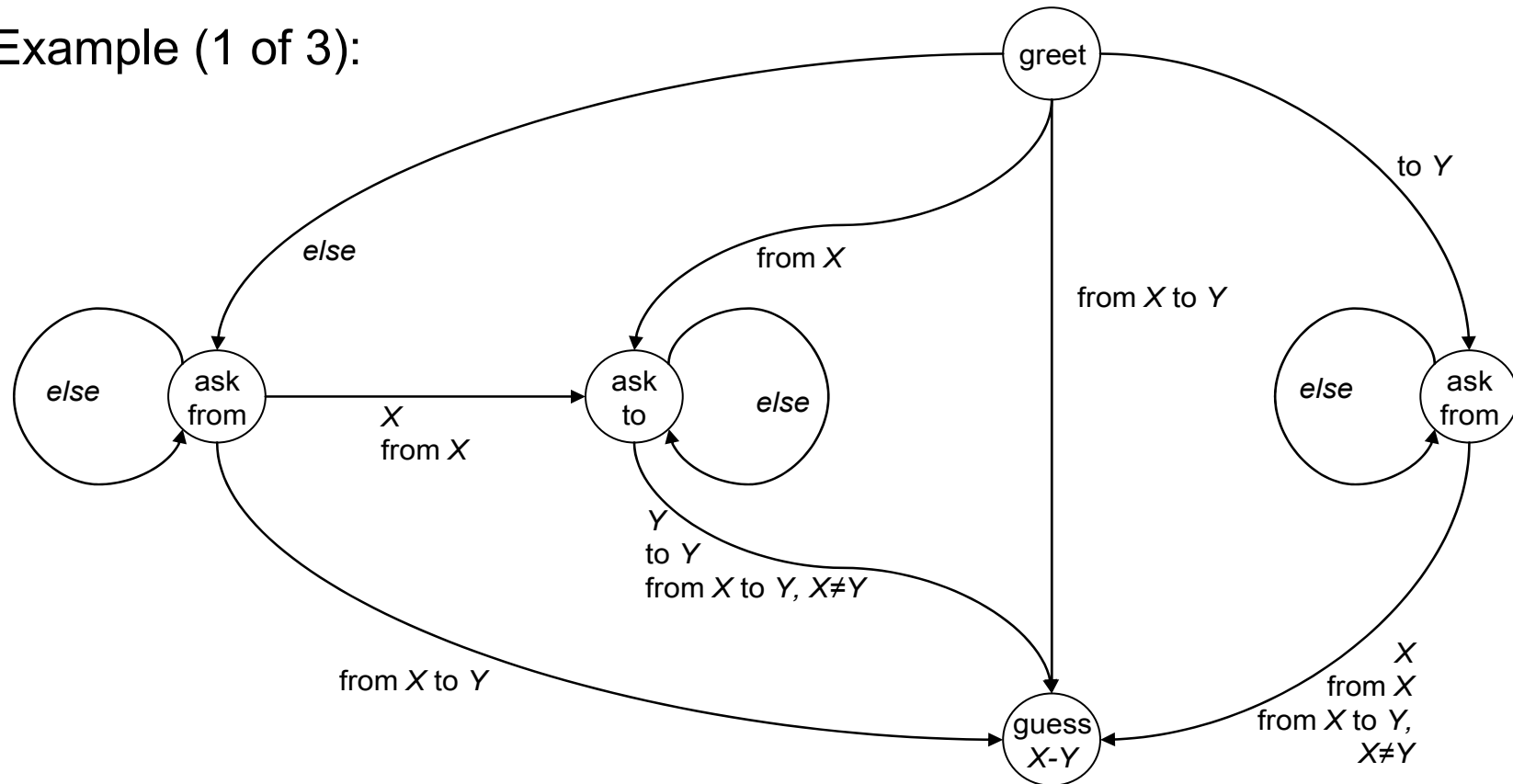




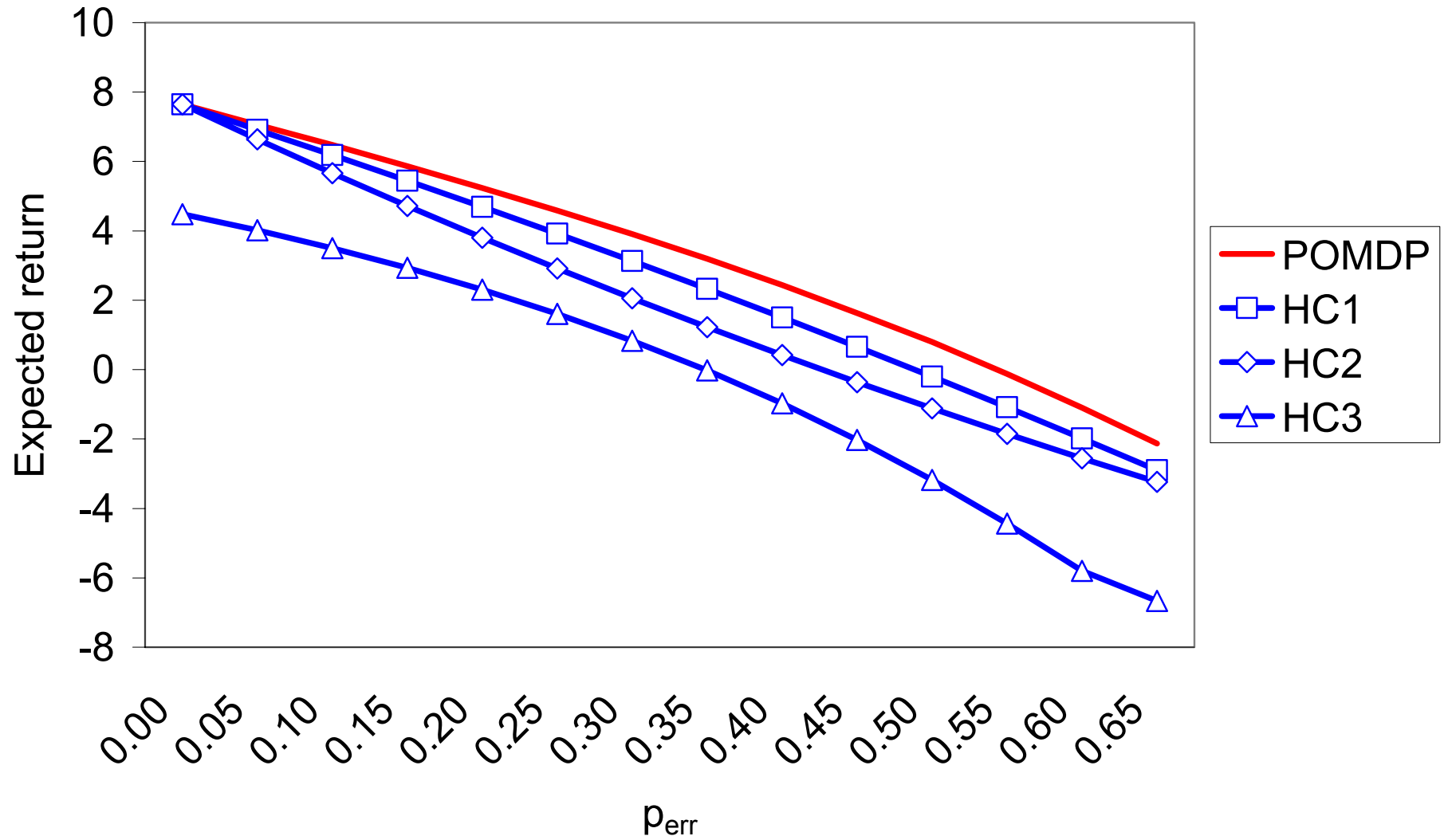
# Baseline 2: handcrafted policies

- Another way to represent a policy: directed graph (FSA)
- We can evaluate the FSA & POMDP to find the expected value of executing the FSA

Example (1 of 3):



# Results: POMDP vs. 3 Handcrafted policies



# Outline

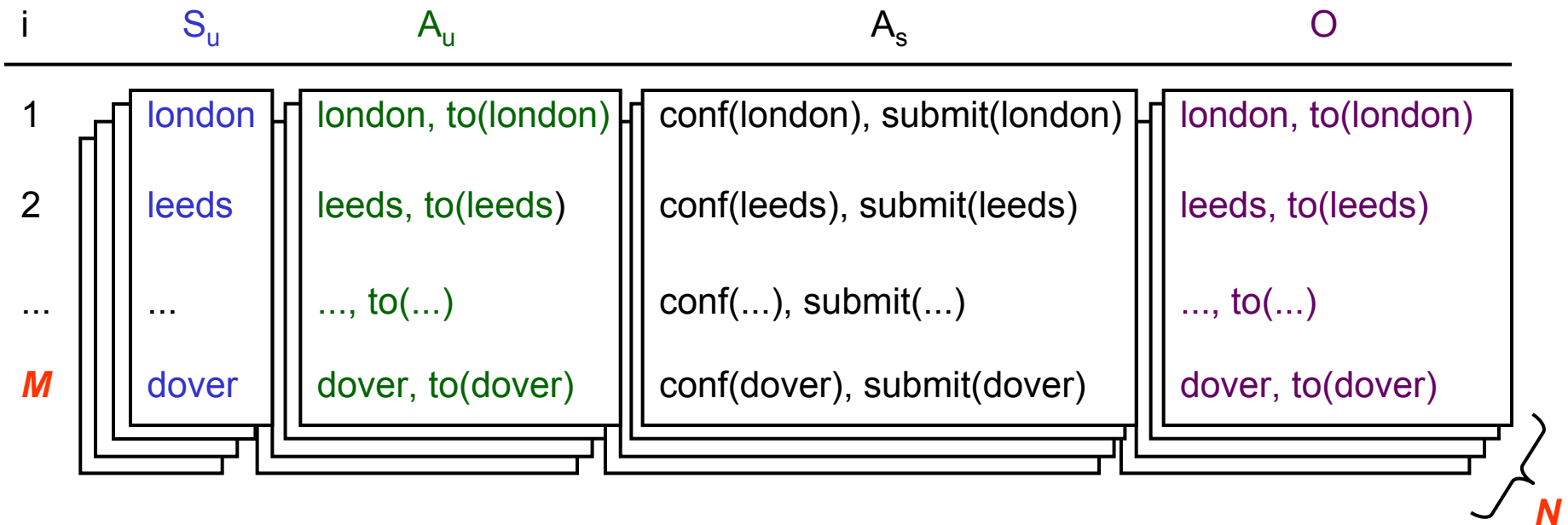


- Motivation
- Model: Proposed POMDP representation of dialogue
- Part 1: Comparison with baselines
  - Example: 3-place problem
- Part 2: Scaling up with the “summary POMDP” method
  - Example: 1000-place problem
- Conclusions & future work



# POMDPs scale poorly

Belief space, actions, and observations cover  $N$  slots and  $M$  slot values



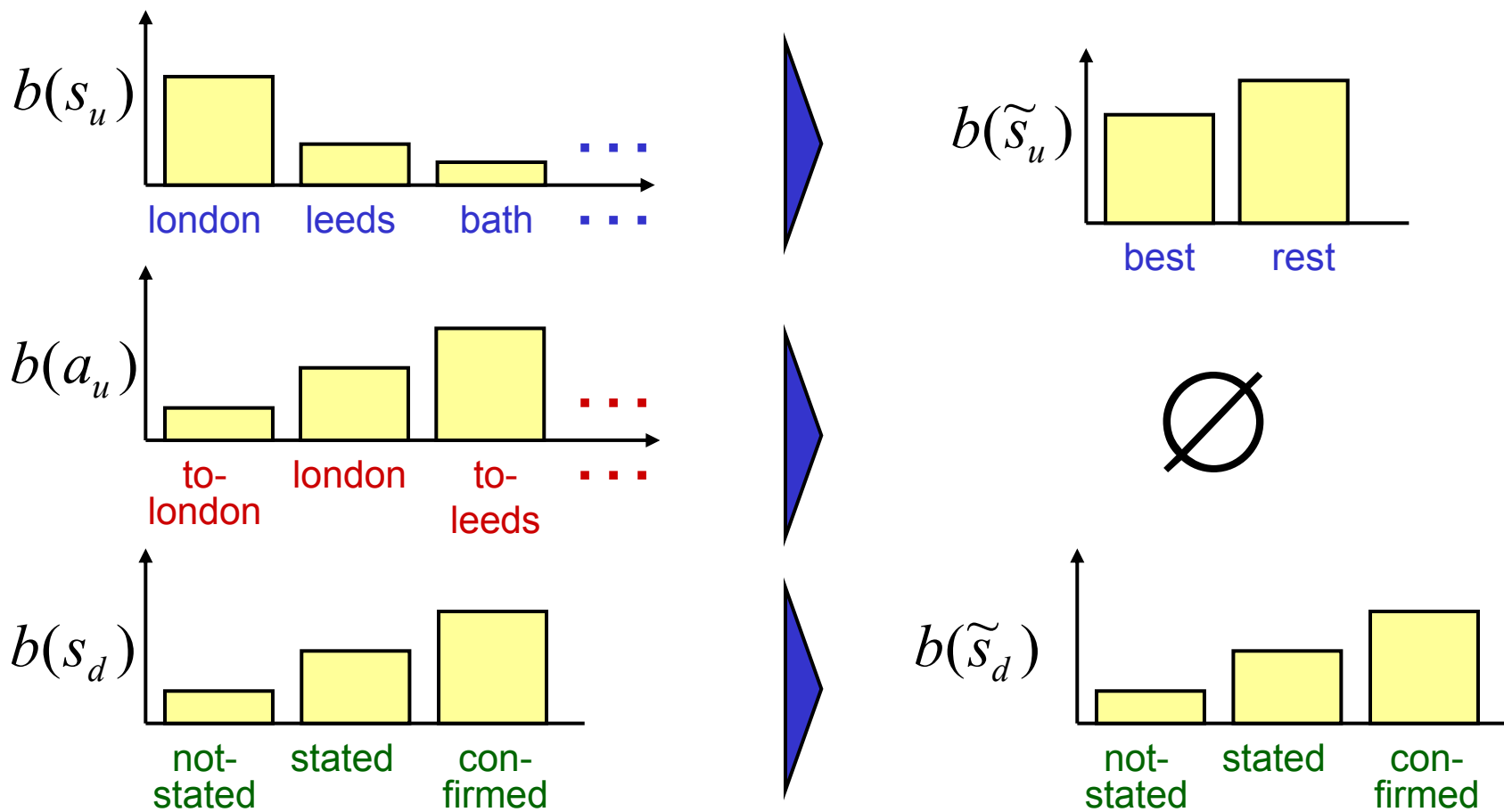
For  $M=1000$  and  $N=2$ , there are  $\sim 10^6$  states, actions, and observations

**Growth factor is  $O(M^N)$**

Can perform belief monitoring with factoring, but how to optimize?



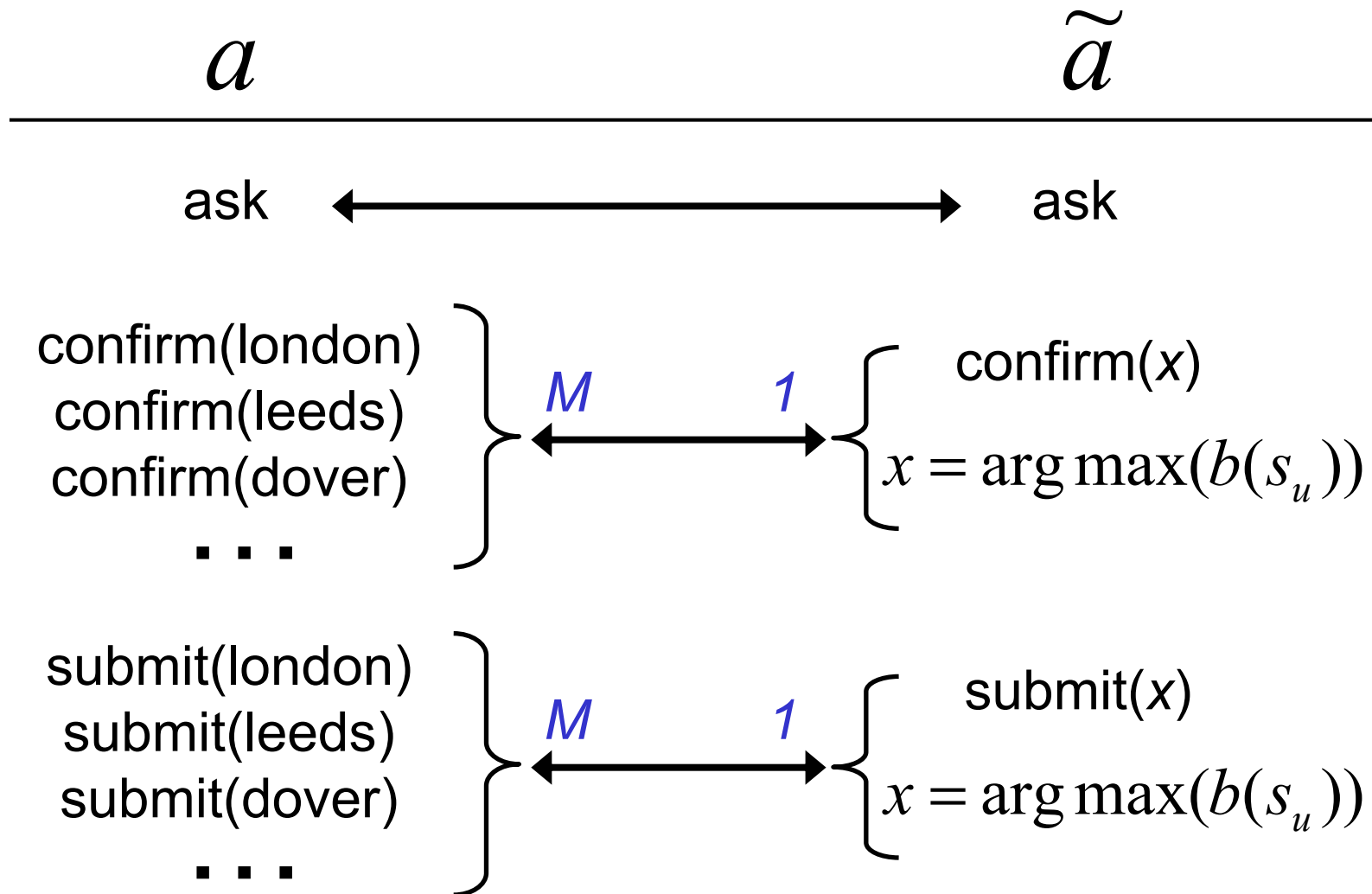
# Intuition of the “Summary POMDP”



$$|S_u| \cdot |A_u| \cdot |S_d| = 1000 \cdot 2000 \cdot 3$$

$$|\tilde{S}_u| \cdot |\tilde{S}_d| = 2 \cdot 3$$

# Actions in the summary POMDP

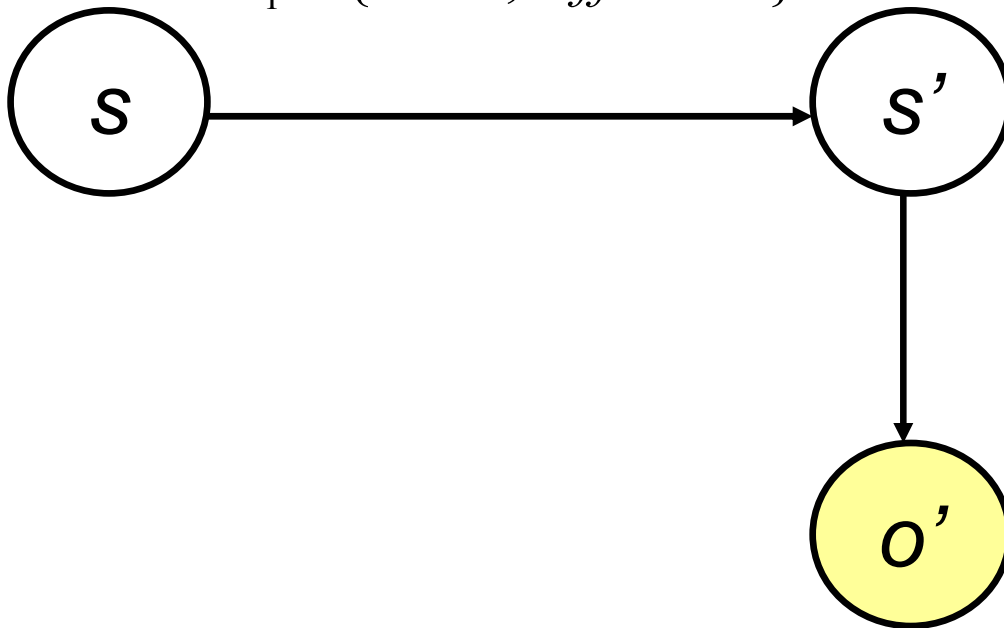




# Observations in the summary POMDP

Observations in summary POMDP give an compact indication of how  $b(\tilde{s})$  is changing based on  $\mathbf{s}$  and  $\mathbf{o}$ . It has 2 parts.

$$\arg \max(b(s_u)) \stackrel{?}{=} \arg \max(b(s'_u))?$$
$$\tilde{o}_1 \in \{same, different\}$$



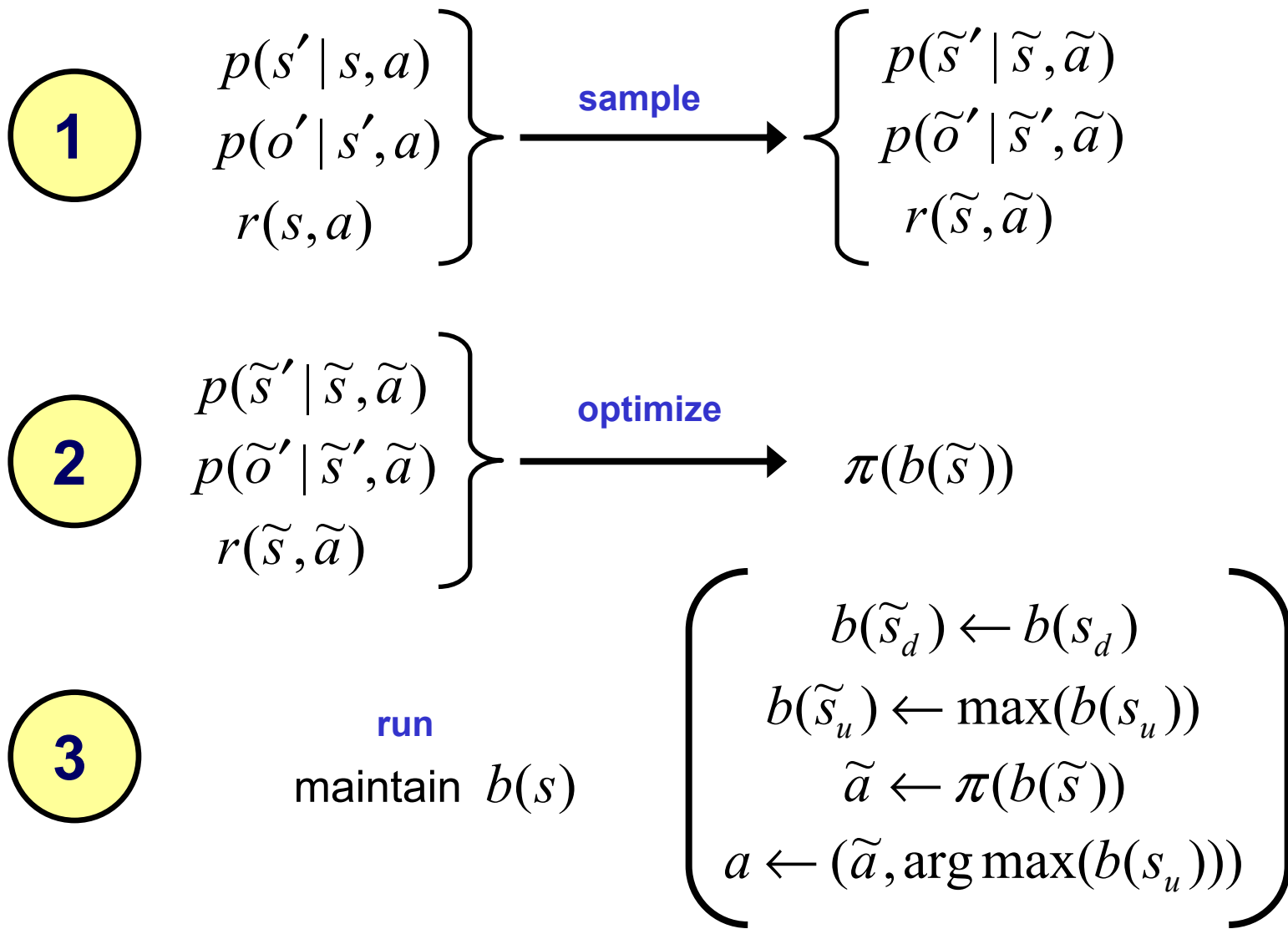
Is  $\arg \max(b(s_u))$

“consistent” with  $\mathbf{o}$ ?

$$\tilde{o}_2 \in \begin{cases} consistent \\ inconsistent \\ no-info \end{cases}$$



# “Summary POMDP” method





# 1000-place problem

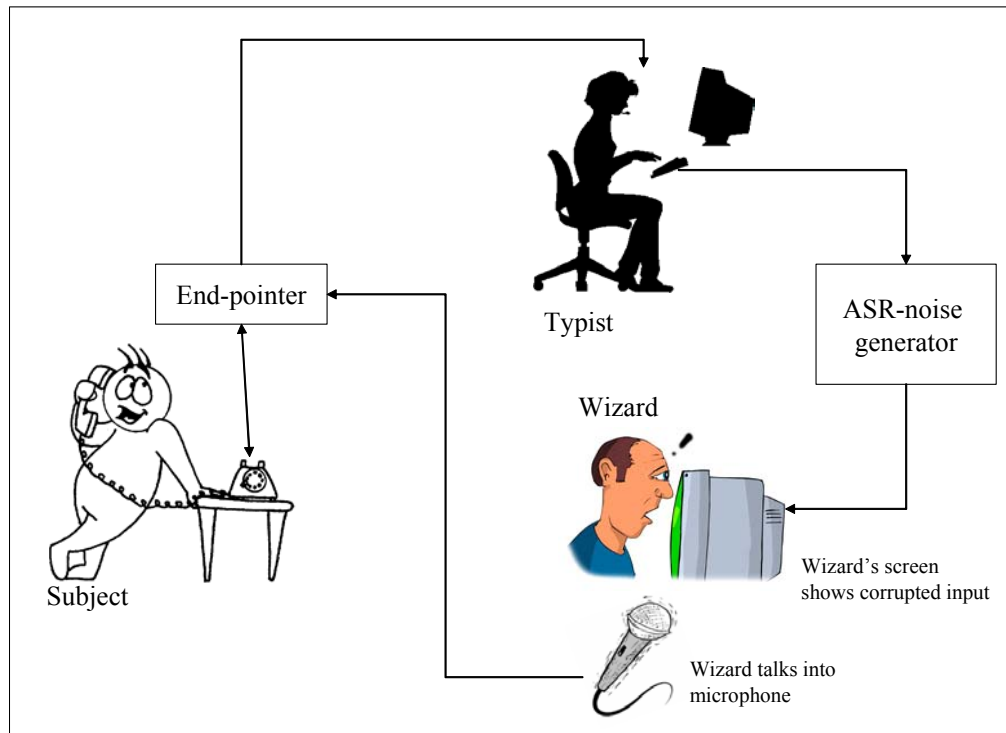
User is travelling from  $x$  to  $y$  in a world with *1000* cities

Same as 3-place problem, with the following extensions

- More factoring, to support efficient belief monitoring
- $a_u$ : can include more complex actions like  $\{yes, to(london)\}$
- **Speech recognition model**: Substitutions & deletions with  $p_{err}$
- **Reward fn**: Richer set of rewards for appropriateness
- **User action model**: Estimated from real dialog data (next)



# Simulated speech recognition channel



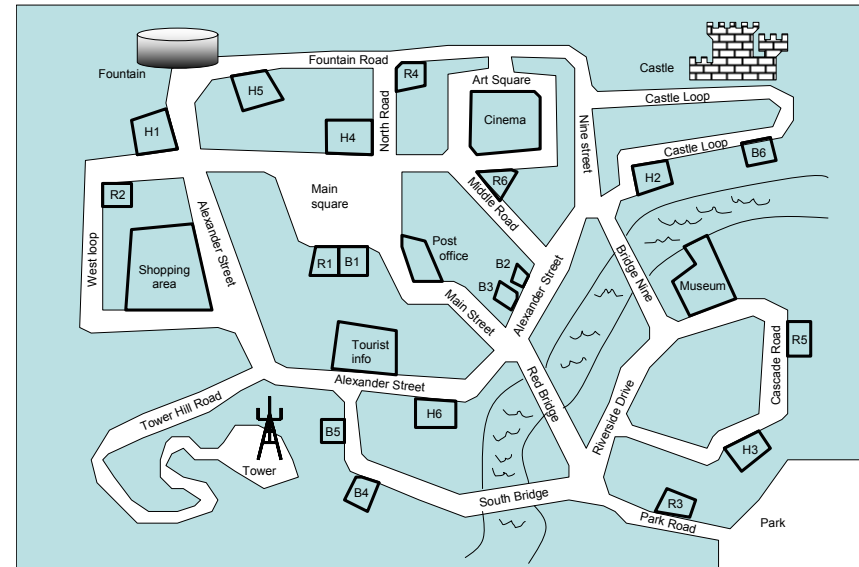
- 2 subjects interact through a *simulated* speech recognition channel
- Typical end-pointing model
- Utterances are transcribed by a typist
- Confusions generated using lexicon, phonetic confusion matrix, and language model

Matthew Stuttle, Jason D. Williams, and Steve Young. *A Framework for Wizard-of-Oz Experiments with a Simulated ASR-Channel*. ICSLP, 2004, South Korea.



# “SACTI-1” dialog data corpus

- “Tourist information” domain
- Wizard has more info than user; user given specific task
- 36 users / 12 wizards
- 144 dialogs
- 4071 turns
- 4 different WER targets
- Task completion
- Likert-scale questions

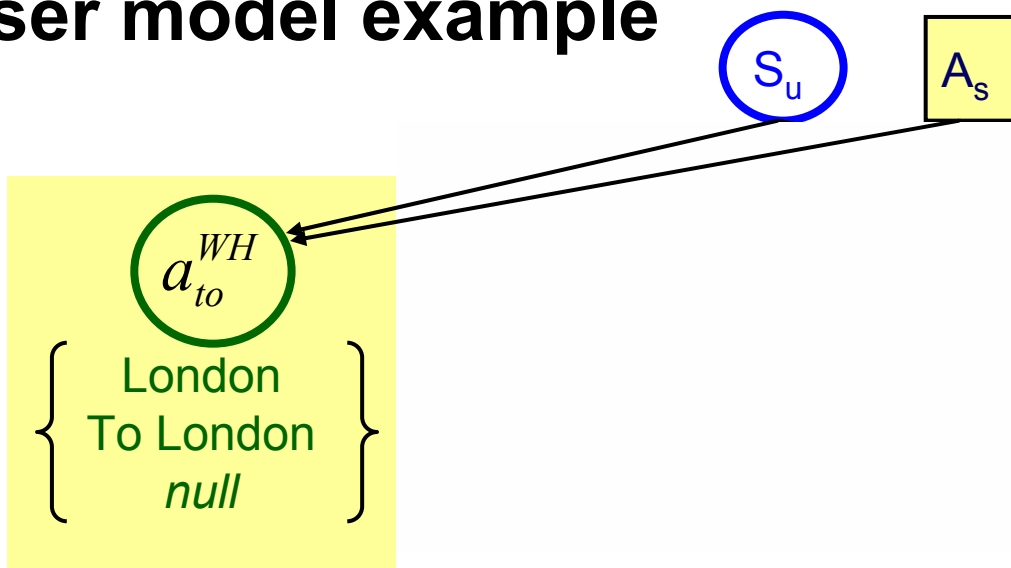


**Conversational characteristics broadly similar to those observed in human/machine dialog.**

**Jason D. Williams and Steve Young. *Characterizing Task-Oriented Dialog using a Simulated ASR Channel*. ICSLP, 2004, South Korea.**



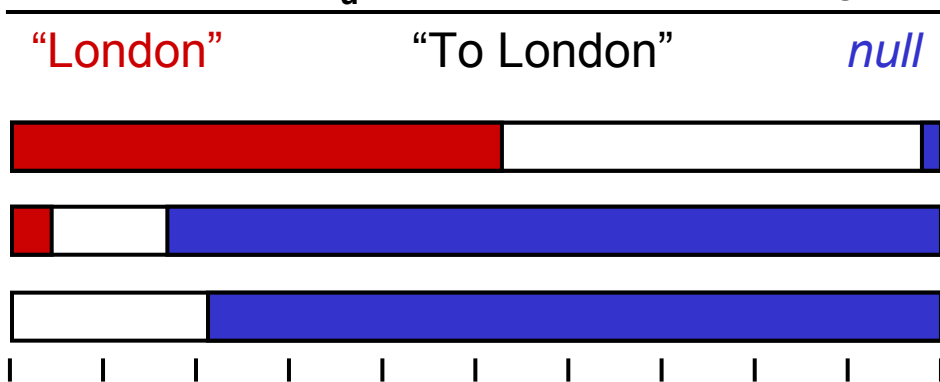
# User model example



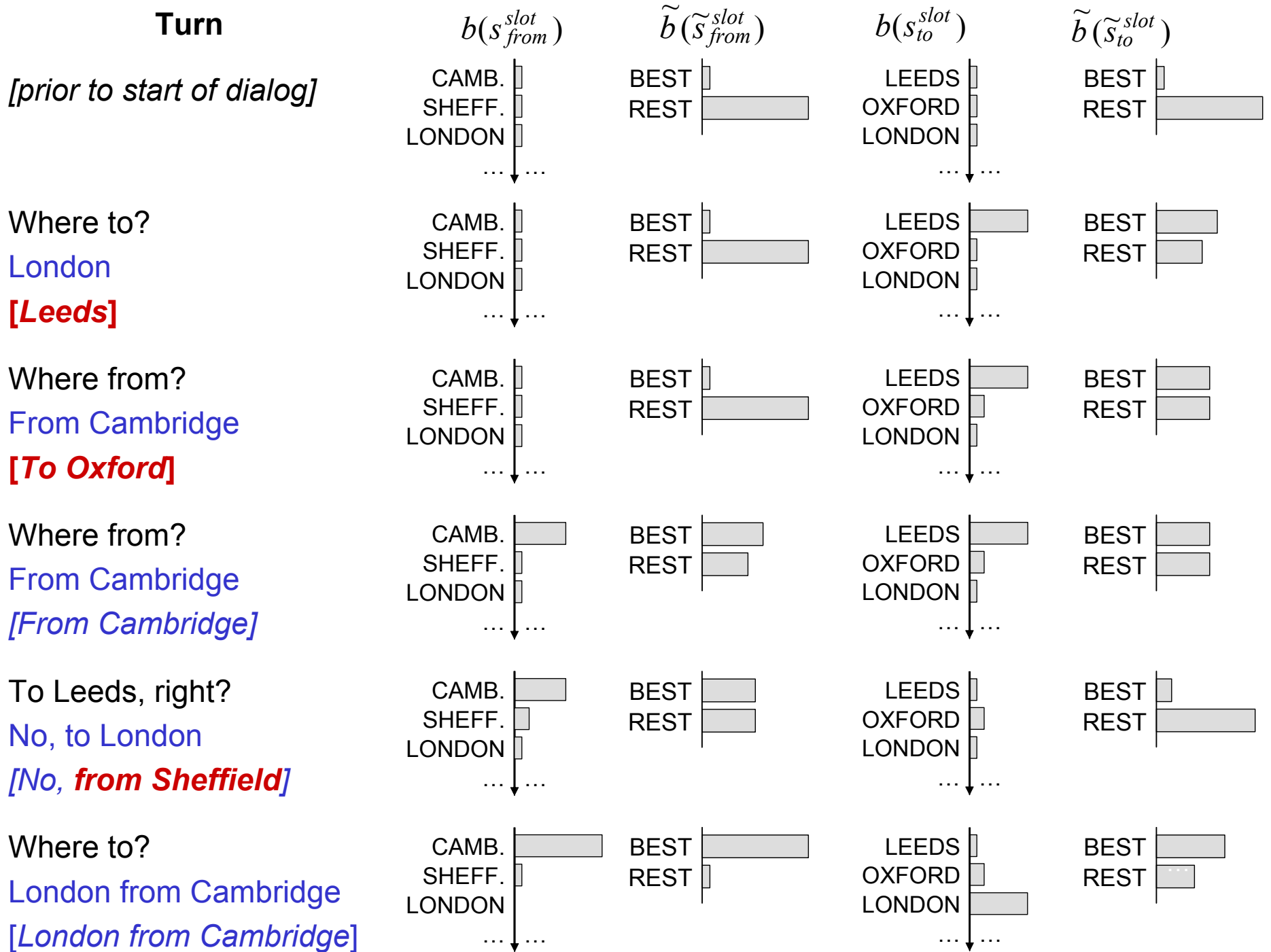
$$P(A_u\text{-}WH \mid A_s, S_u=\text{to}(\text{london}))$$

| $A_s$                     |
|---------------------------|
| Where are you going to?   |
| To London, is that right? |
| Where are you going from? |

P mass of  $A_u$  ("to/WH" portion only)

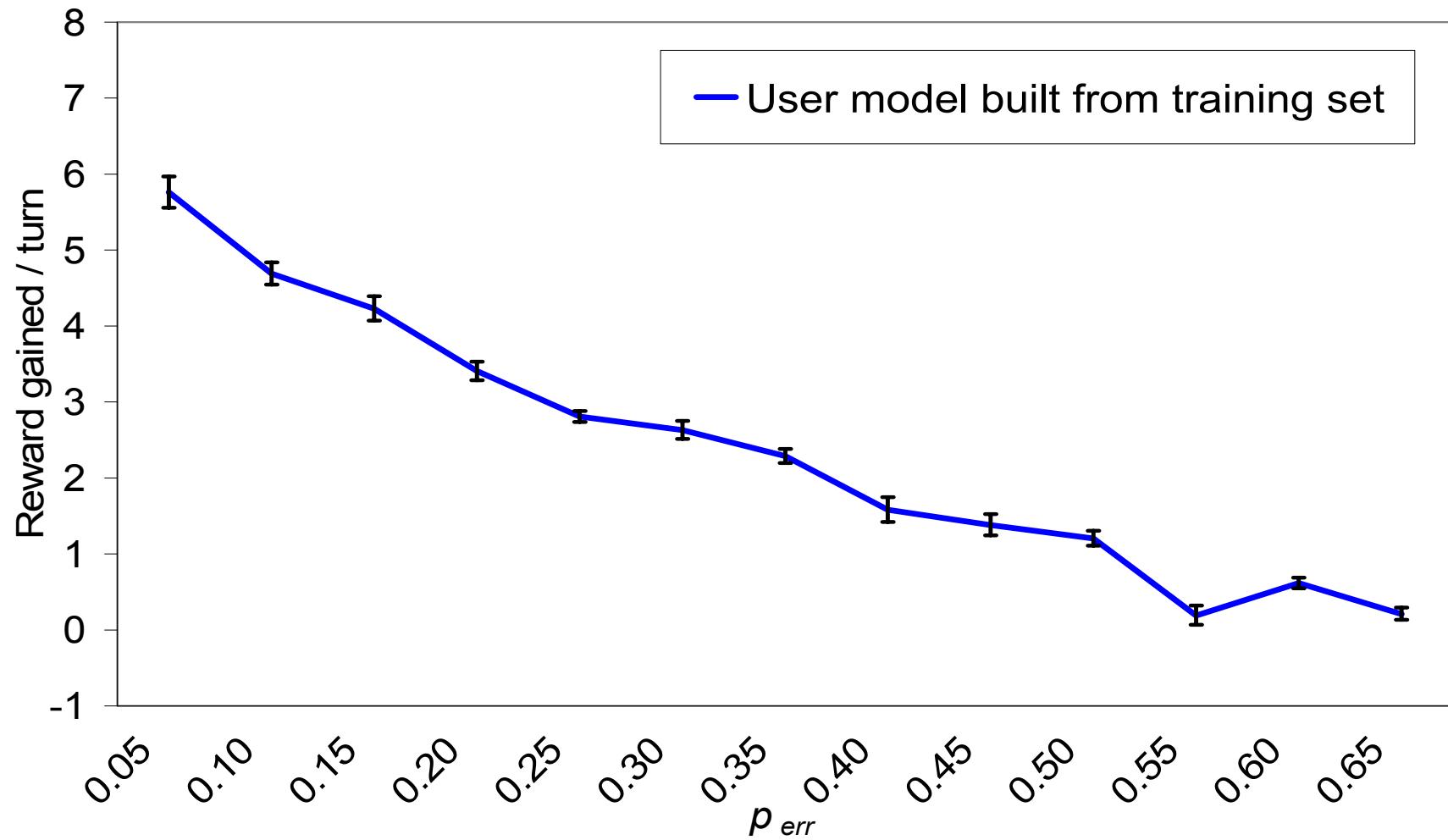


**Data was divided, and two model sets were created:  
Training model set & Testing model set**



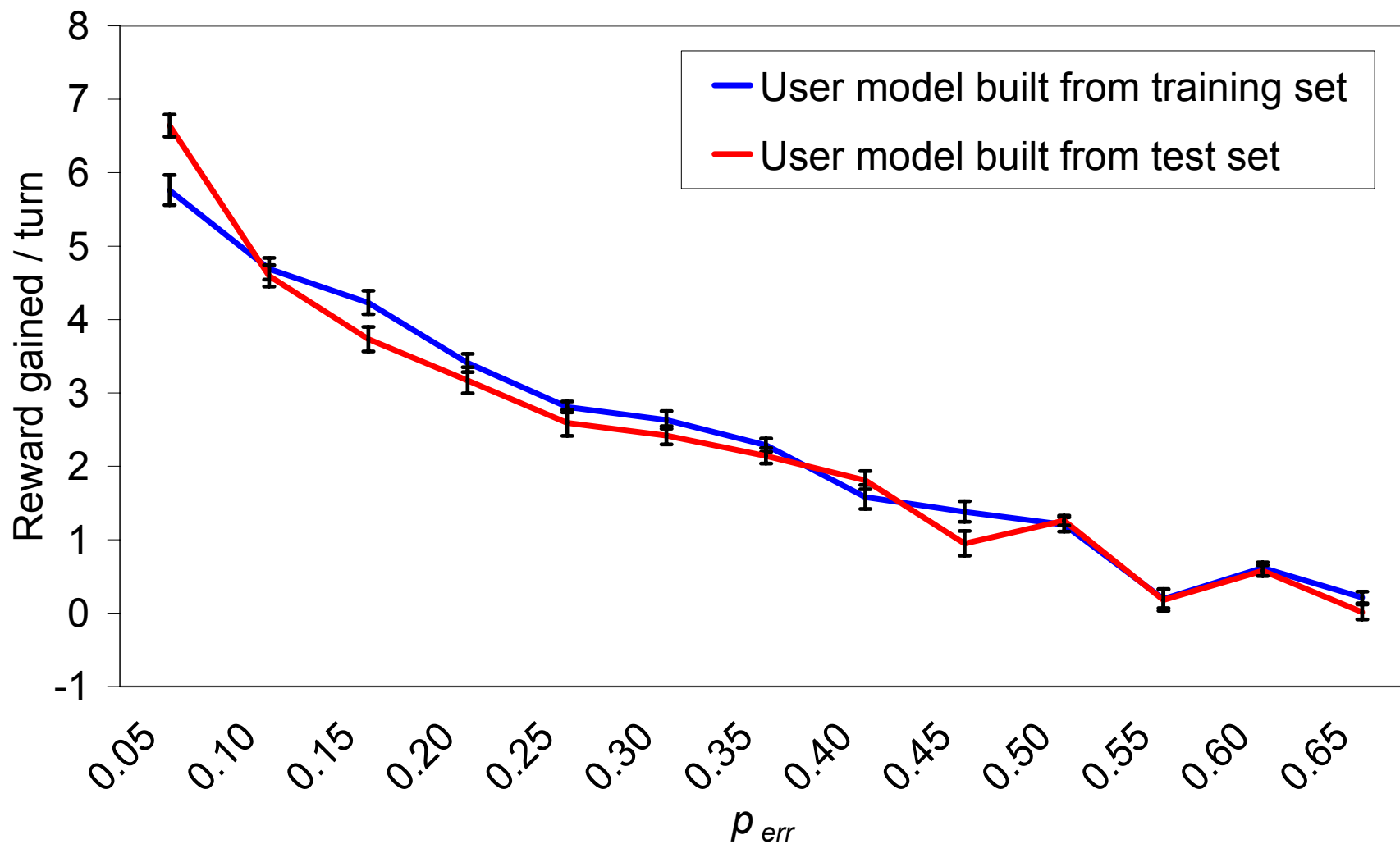


# Performance vs. $p_{err}$



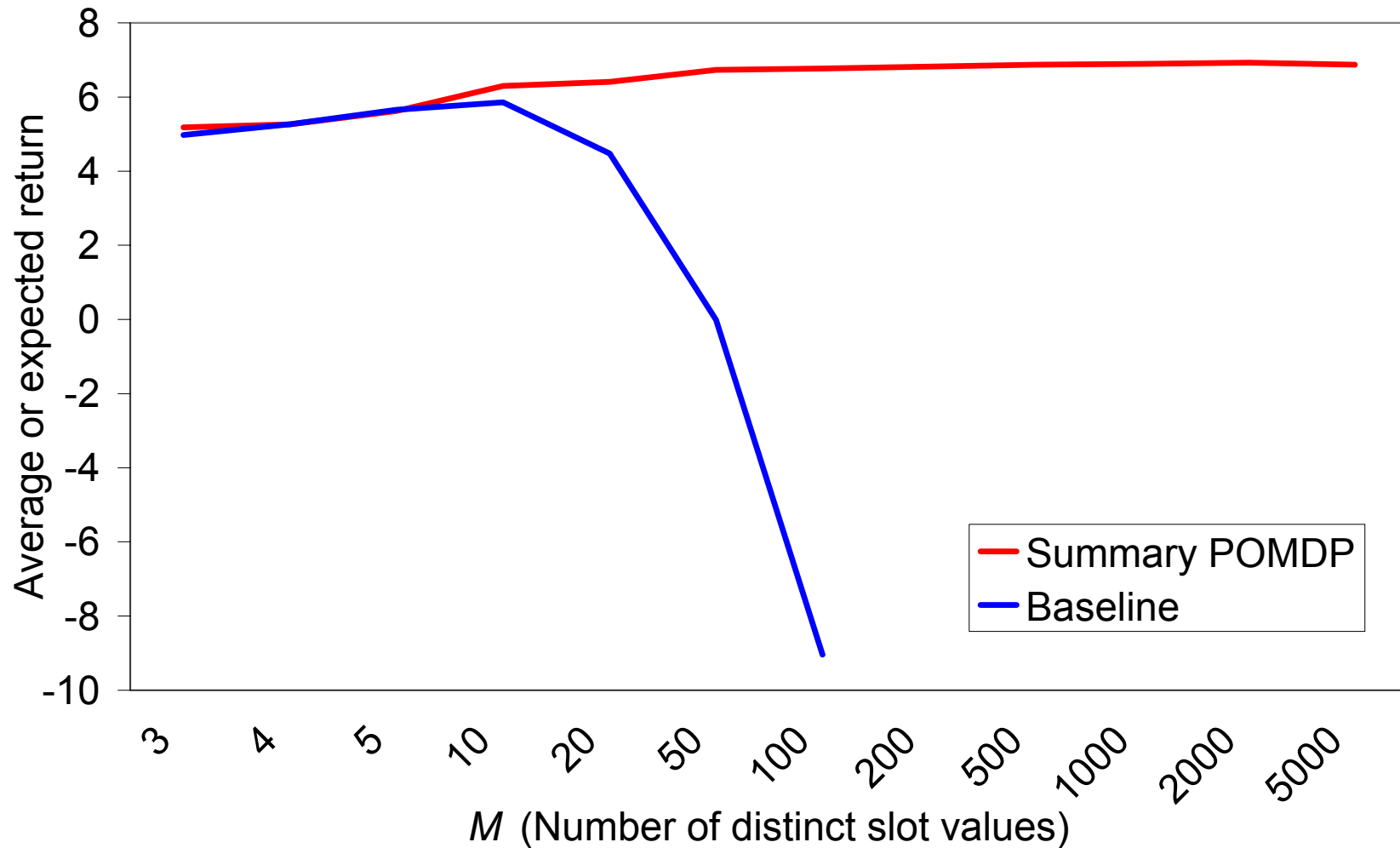


# Performance vs. $p_{err}$





# Scalability vs. direct optimization



# Outline



- Motivation
- Model: Proposed POMDP representation of dialogue
- Part 1: Comparison with baselines
  - Example: 3-place problem
- Part 2: Scaling up with the “summary POMDP” method
  - Example: 1000-place problem
- **Conclusions & future work**



# Conclusions & future work

- For slot-based dialog management, POMDPs...
  - Outperform MDP baseline
  - Outperform 3 handcrafted baselines
  - Can be scaled as a “summary POMDP” to many slot values
- Interesting future avenues
  - Scale *number of* slots to 10s
  - Move beyond slot-based approaches
  - Apply to real system

# Thanks!



*Jason D. Williams*

jdw30@cam.ac.uk

Joint work with

*Pascal Poupart*

ppoupart@cs.waterloo.ca

*Steve Young*

sjy@eng.cam.ac.uk

