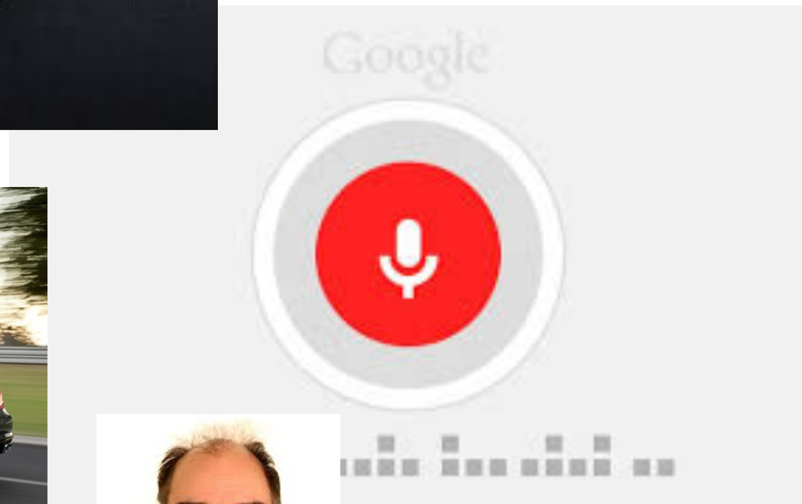
An introduction to Statistical Spoken Dialogue Systems

UK Speech 2013

Presented by Blaise Thomson

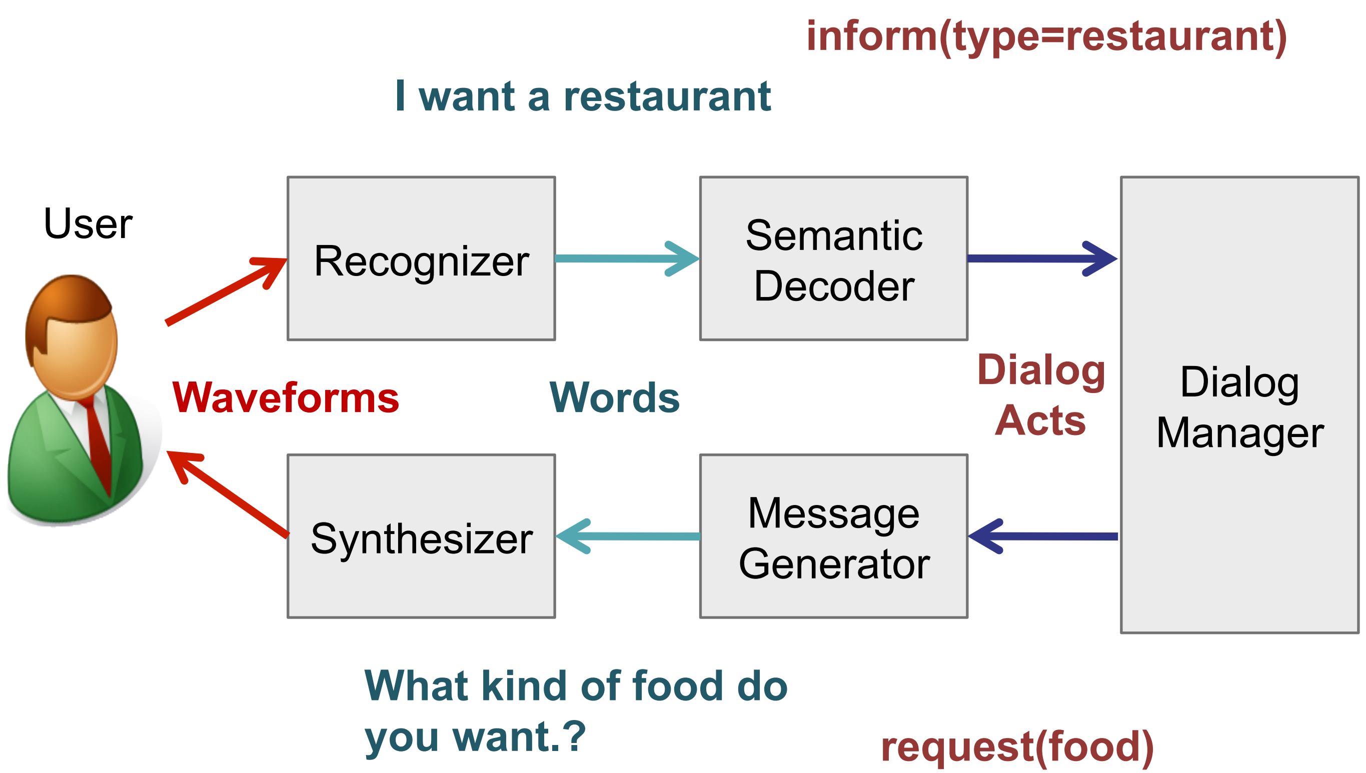
Cambridge University Engineering Department
brmt2@eng.cam.ac.uk
http://mi.eng.cam.ac.uk/~brmt2

Spoken Dialogue Systems – Examples



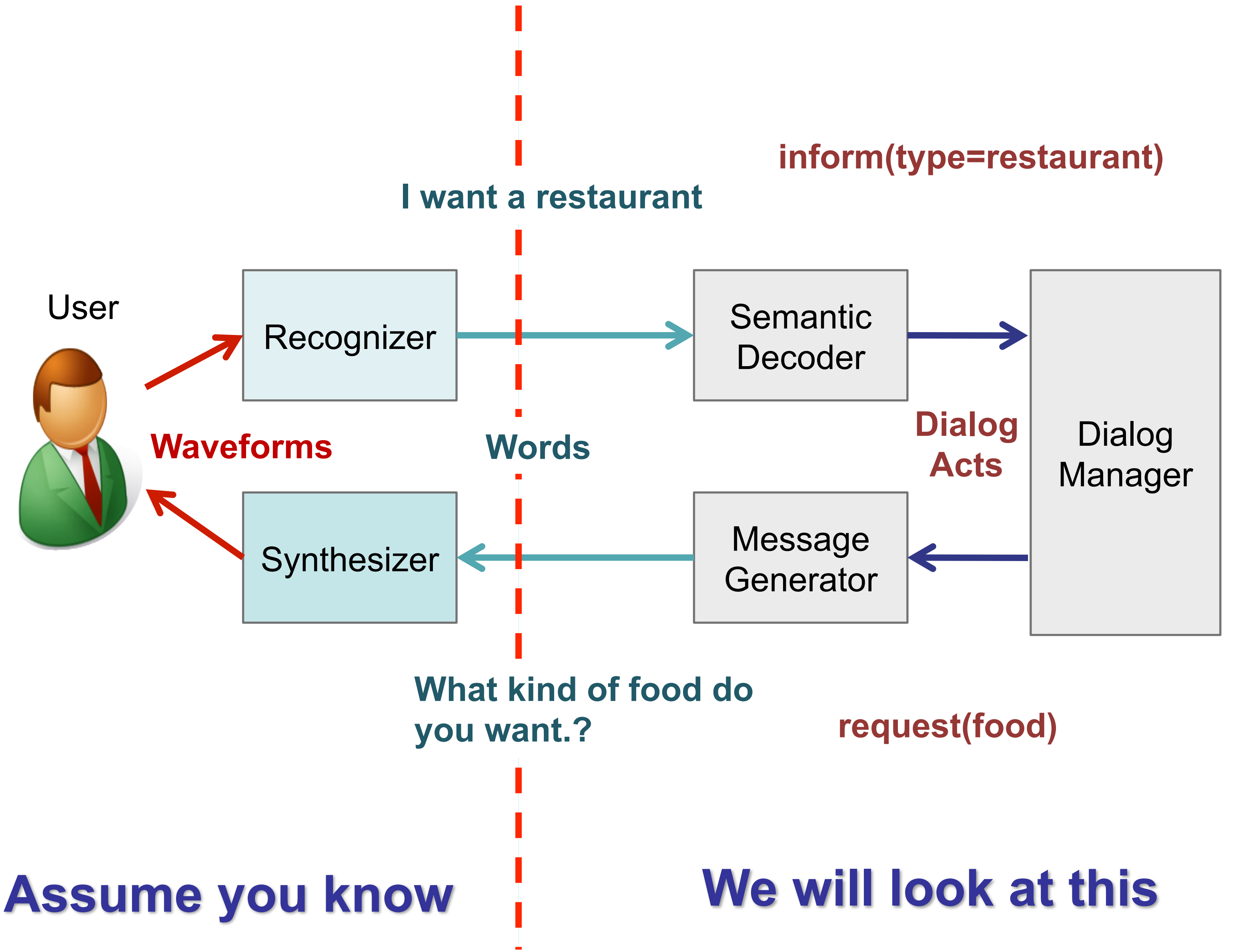
Cambridge system demo - 01223 852 453

Human-machine spoken dialogue



Typical structure of a spoken dialogue system

Human-machine spoken dialogue



Semantic decoding - Intro

Is there um maybe a cheap place in the centre of town please?

inform (price = cheap, area = centre)

dialogue act type

semantics slots and values

- Lots of disfluencies in speech – grammars tend to break
- We don’t care about the exact meaning
  - We just want to know what the user wants
  - Idea of speech act / dialog act (Austin / Searle / Traum)

Semantic decoding – a simple approach

Is there um maybe a cheap place in the centre of town please?

Word / Bigram	x_i
is	1
there	1
yes	0
um	1
…	
is there	1
is you	0
…	

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ \ldots \\ 1 \\ 0 \\ \ldots \end{pmatrix}$$

Semantic decoding – a simple approach

inform (price = cheap, area = centre)

Act type
inform

Assign type number (e.g. 3)

Item	y_i
price=cheap	1
price=exp	0
…	…
area=centre	1
area=west	0
…	…

$$\begin{pmatrix} 1 \\ 0 \\ \dots \\ 1 \\ 0 \\ \dots \end{pmatrix}$$

Semantic decoding – a simple approach

- When training we have lots of input vectors $\mathbf{x_t}$ and output vectors $\mathbf{y_t}$

- Use your favourite supervised learning algorithm
  - Naïve Bayes
  - Logistic regression
  - Support Vector Machines (Mairesse et al, 2009)
  - Others?

- Act type is multi-class labeling task, others are all just binary

Semantic decoding – a summary

- Words are inputs
  - Convert them to vectors (1/0)
  - Add bigrams / trigrams
  - Confusion network features too! (Henderson et al 2012)

- Act type + slot values are outputs
  - Convert them to vectors (1/0)

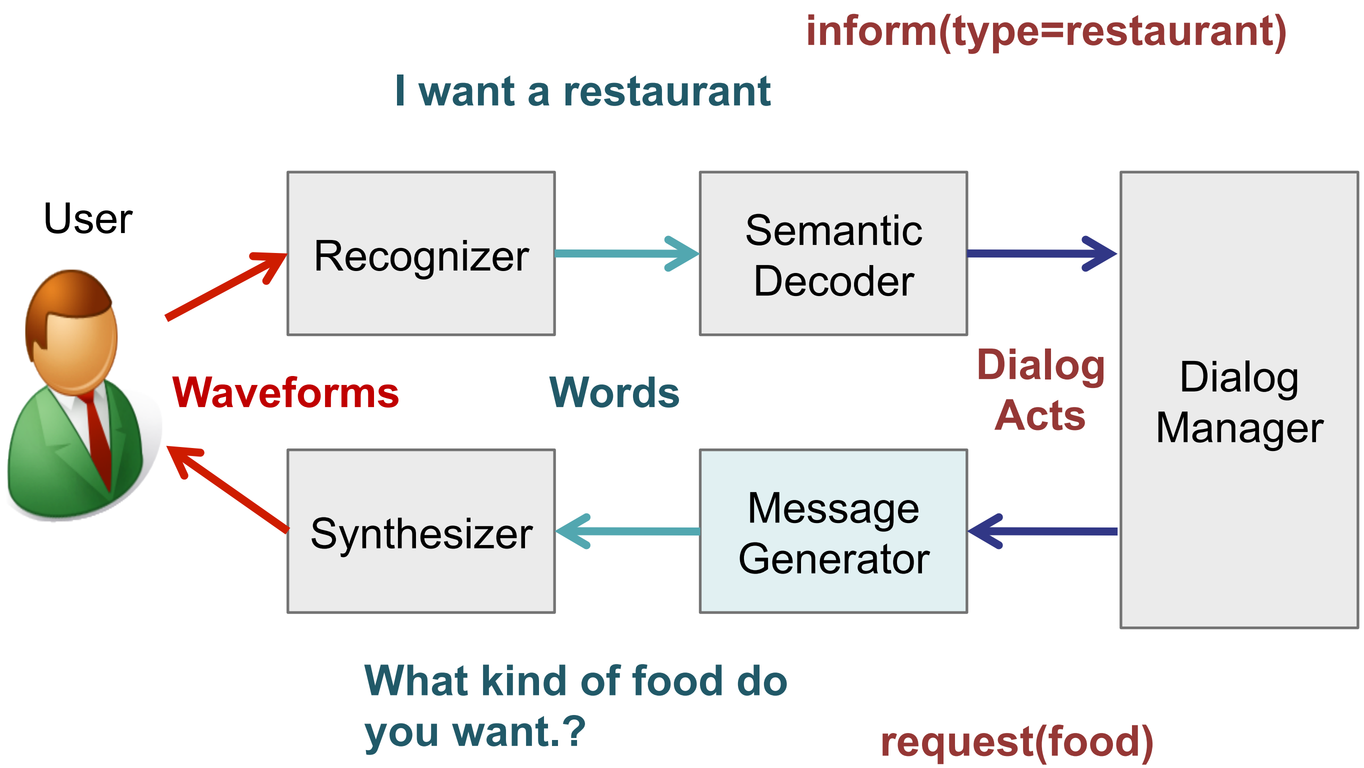
- Run your favourite learning algorithm

- In practice – may help to post-process a bit

Semantic decoding – Other approaches

- Handcrafted grammars
  - Phoenix (Ward et al 1994)
- Hidden Vector State model
  - HMM structure, with hidden stack of concepts
  - He & Young (2005)
- Using Markov Logic Networks
  - Meza-Ruiz (2008)
- Transformation based approach
  - Jurcicek et al (2009)
- Using Combinatory Categorial Grammars (CCG)
  - Supervised - Zettlemoyer & Collins (2009)
  - Unsupervised - Artzi & Zettlemoyer (2011)

Output generation



Output generation - Templates

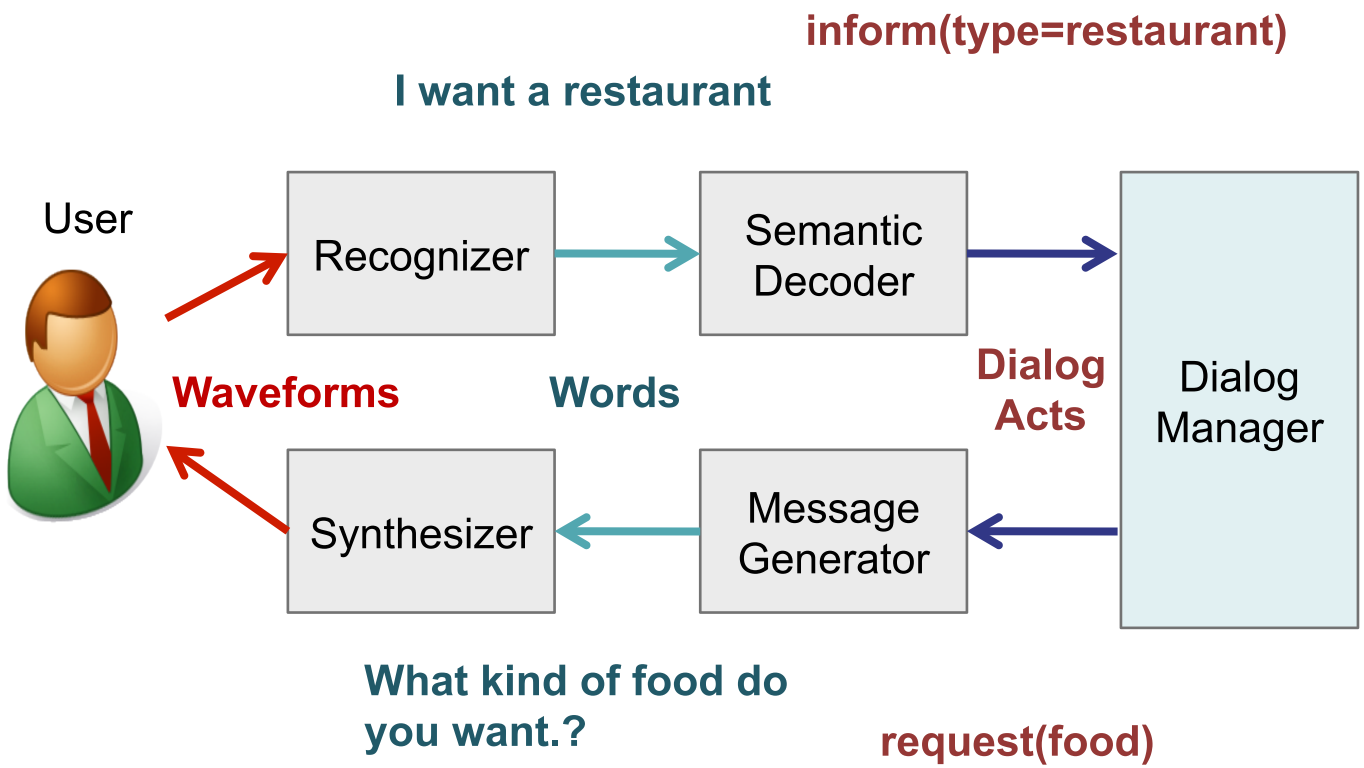
- To generate natural sentences, many systems use templates

  inform(name=$X, area=$Y) => “$X is in the $Y of town”

  inform(name=“Char Sue”, area=centre) =>
  “Char Sue is in the centre of town”

- Some work has been done on learning the generator
  - Overgenerate and rank (Langkilde & Knight 1998)
  - Bayesian Networks (Mairesse et al 2010)
  - Conditional Random Fields (Dethlefs et al 2013)

Human-machine spoken dialogue



Dialogue management – what to say?

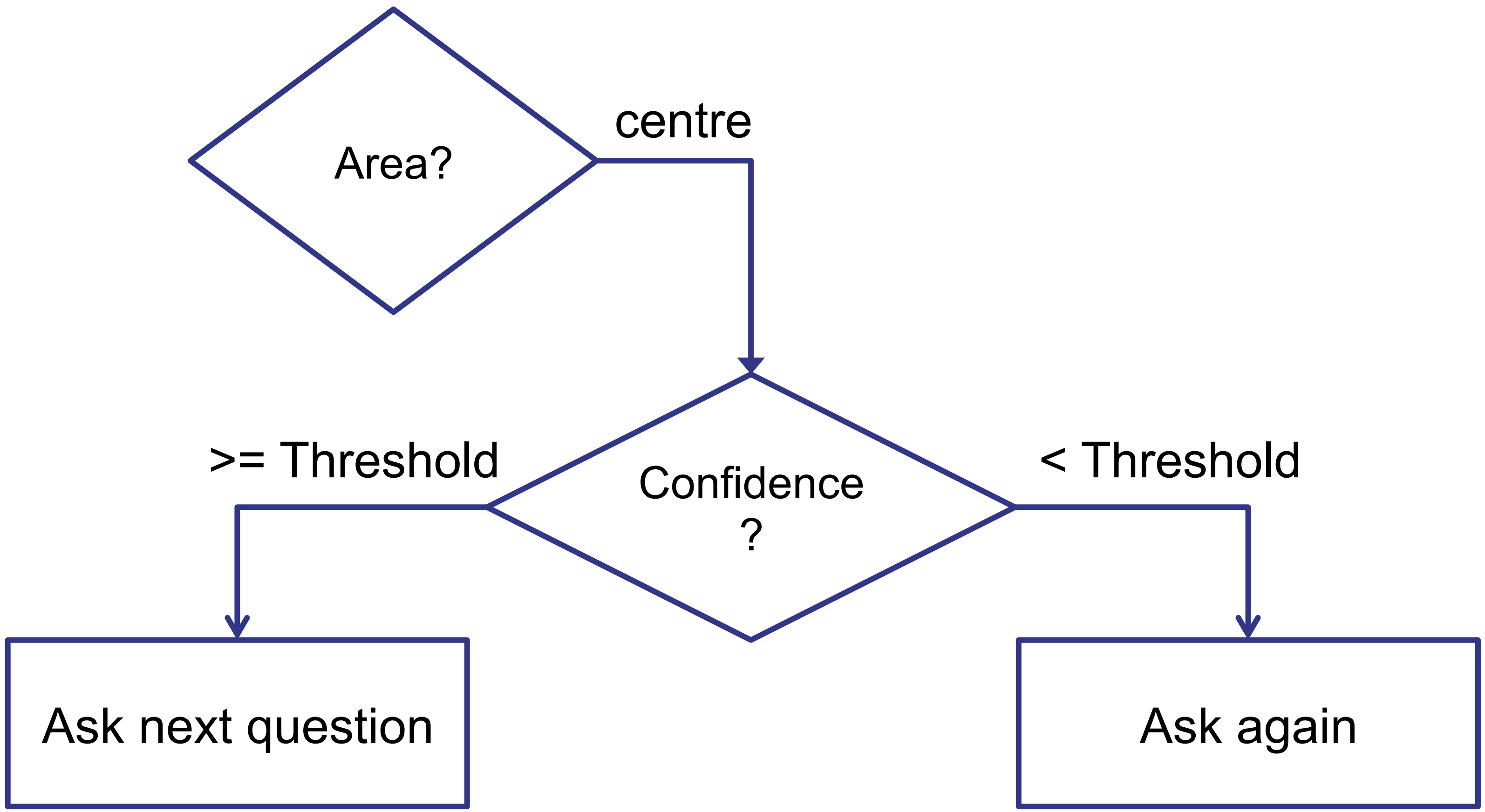
State model

(where are we)

Policy model

(what to do)

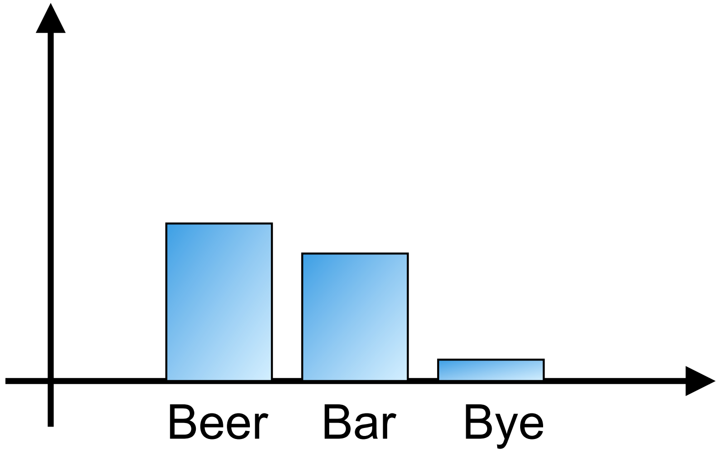
State model – The traditional approach



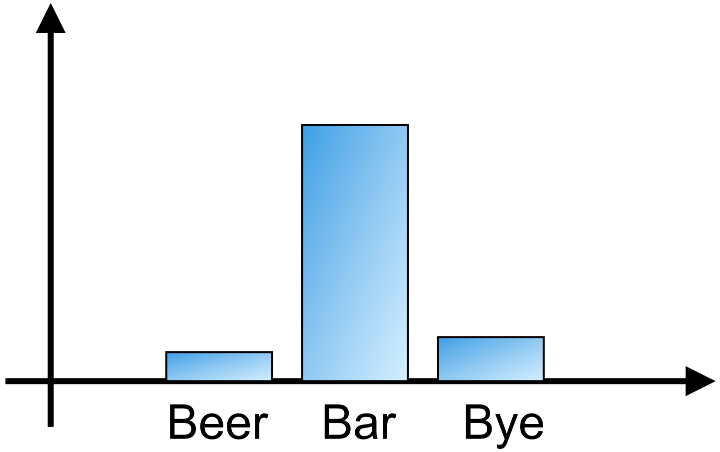
State model – The traditional approach

State model – The probabilistic approach

- How can I help you?
  - I’m looking for a beer [0.5]
  - I’m looking for a bar [0.4]



- Sorry, what did you say?
  - bar [0.3]
  - bye [0.3]



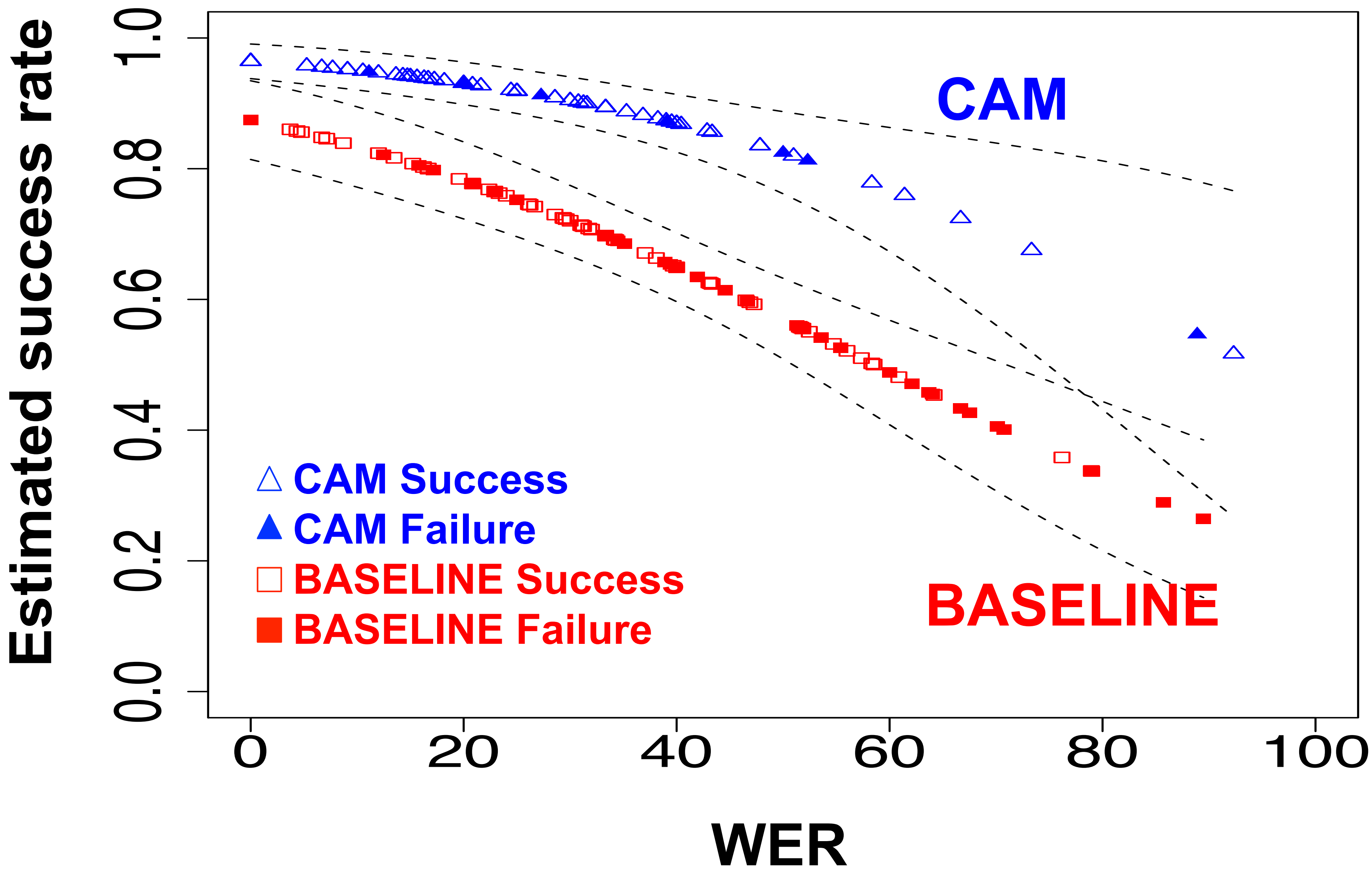
State model – probabilities help

- Tested in the Spoken Dialogue Challenge (2009)
- Provide bus timetables in Pittsburgh
- 800 road names (pairs represent a stop). Required to get place from, to and time

	# Dial	# Succ	% Succ	WER
BASELINE	91	59	64.8 +/- 5.0	42.35
System 2	61	23	37.7 +/- 6.2	60.66
Probabilistic	75	**67**	**89.3 +/- 3.6**	**32.65**
System 4	83	62	74.7 +/- 4.8	34.34

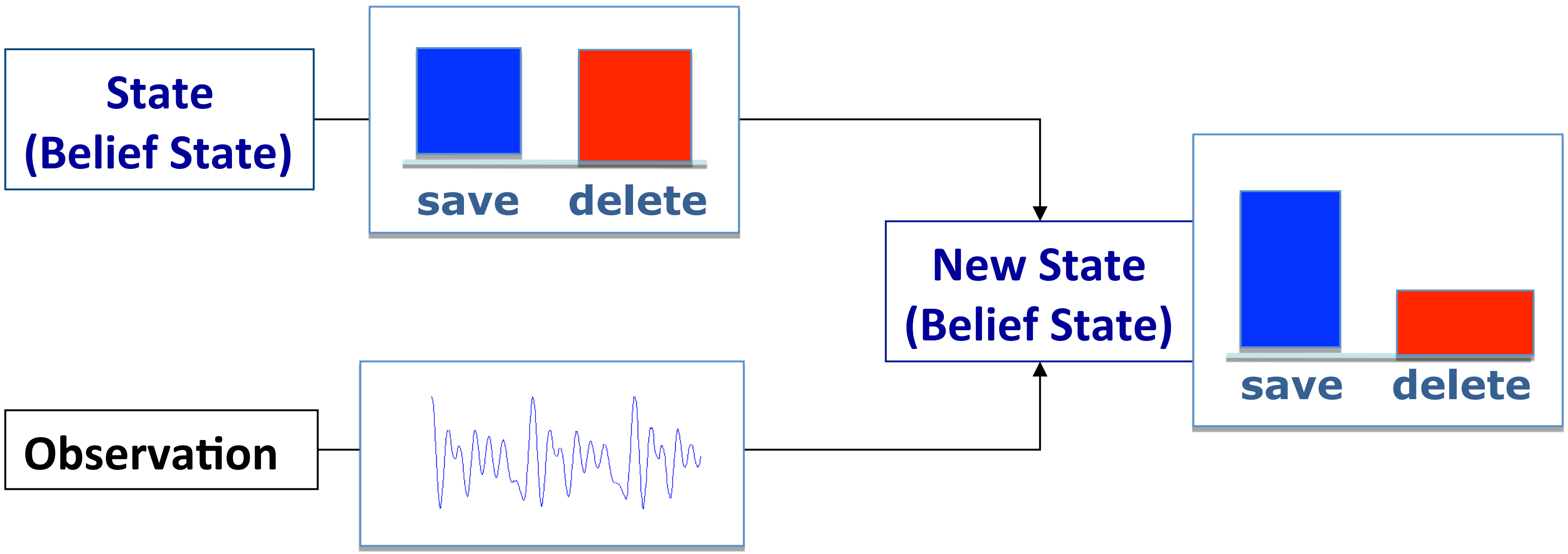
Thomson et al SLT 2010

State model – probabilities help

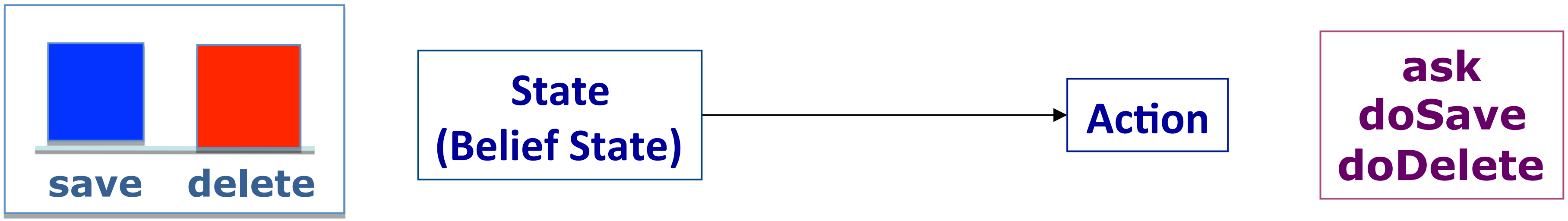


Dialog – what would you do?

1. State tracking



2. Policy decisions



Current approaches

1. State tracking:

- Hand-crafted
- Generative
- Discirimінative

2. Policy decisions:

- Deterministic
  - Hand-crafted via flow-chart
  - Logic representations
- Supervised learning
- Reinforcement learning

Current approaches

1. State tracking:

- Hand-crafted
- Generative
- Discirimiinative

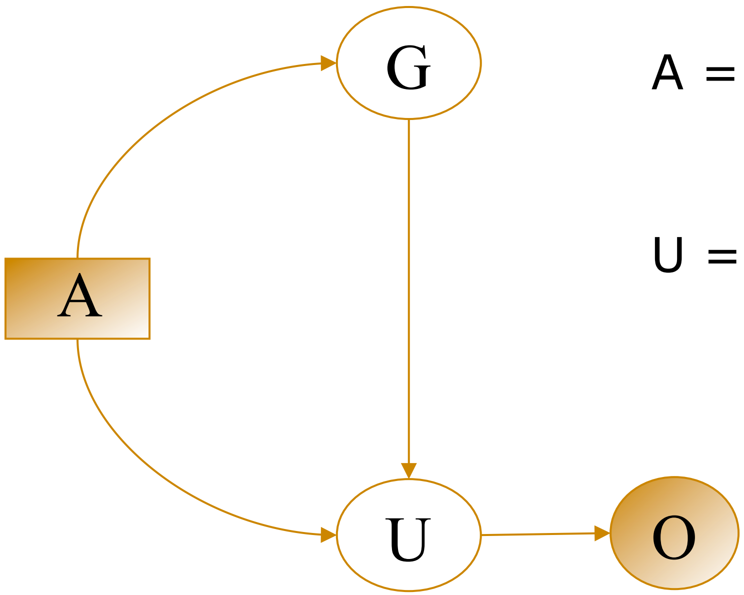
2. Policy decisions:

- Deterministic
  - Hand-crafted via flow-chart
  - Logic representations
- Supervised learning
- Reinforcement learning

Generative state tracking

$$p(G|A,O) = k\sum_U p(O,U,G,A)$$

$$= k\sum_U p(O|U)p(U|G,A)p(G|A)$$

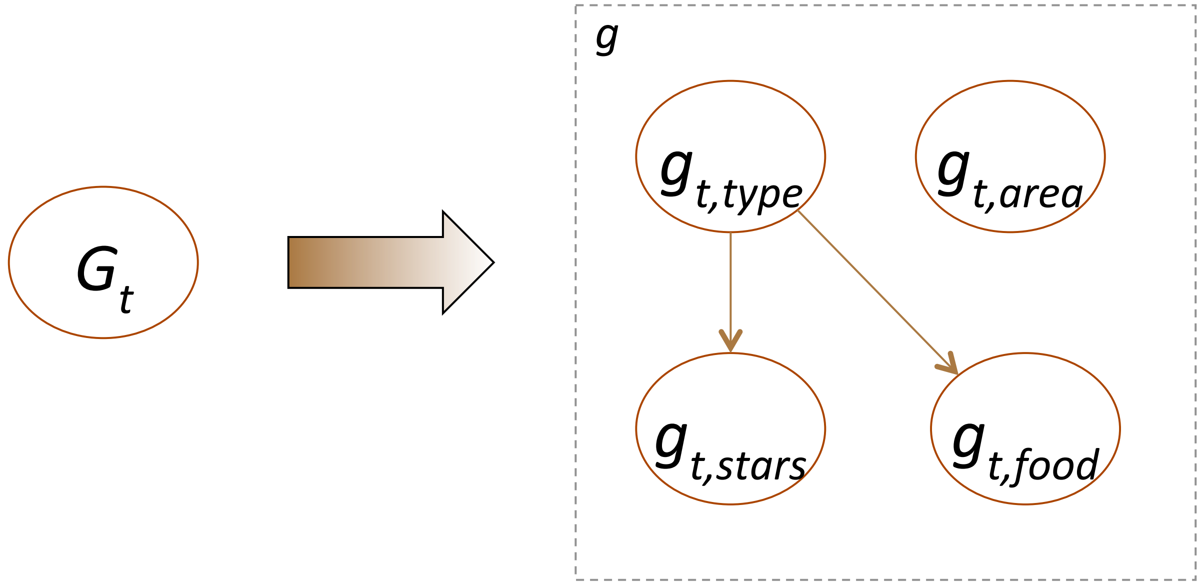


A = System Action O = Observation

U = User Action G = User Goal

Generative state tracking

- Computing joint probabilities is intractable
- Split the goal, G_t, into sub-goals $g_{t,c}$
- Assume sub-goals are conditionally independent
- e.g. User wants a Chinese restaurant → food=Chinese, type=restaurant

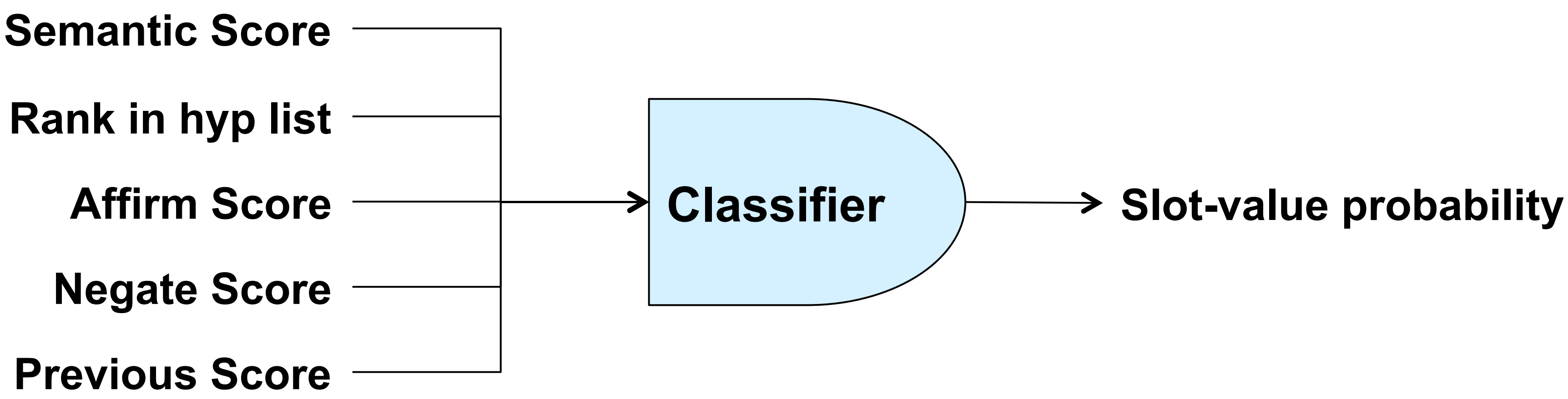


Generative state tracking – summary

- Simplify via Bayesian networks
  - Loopy belief propagation computes the probabilities
  - Expectation propagation computes parameters
- Parameters can be learned unsupervised
- The model can generate
- It is often difficult to include complex dependencies

Discriminative state tracking

- Each slot prediction is a classification task
- Put in whatever features you want

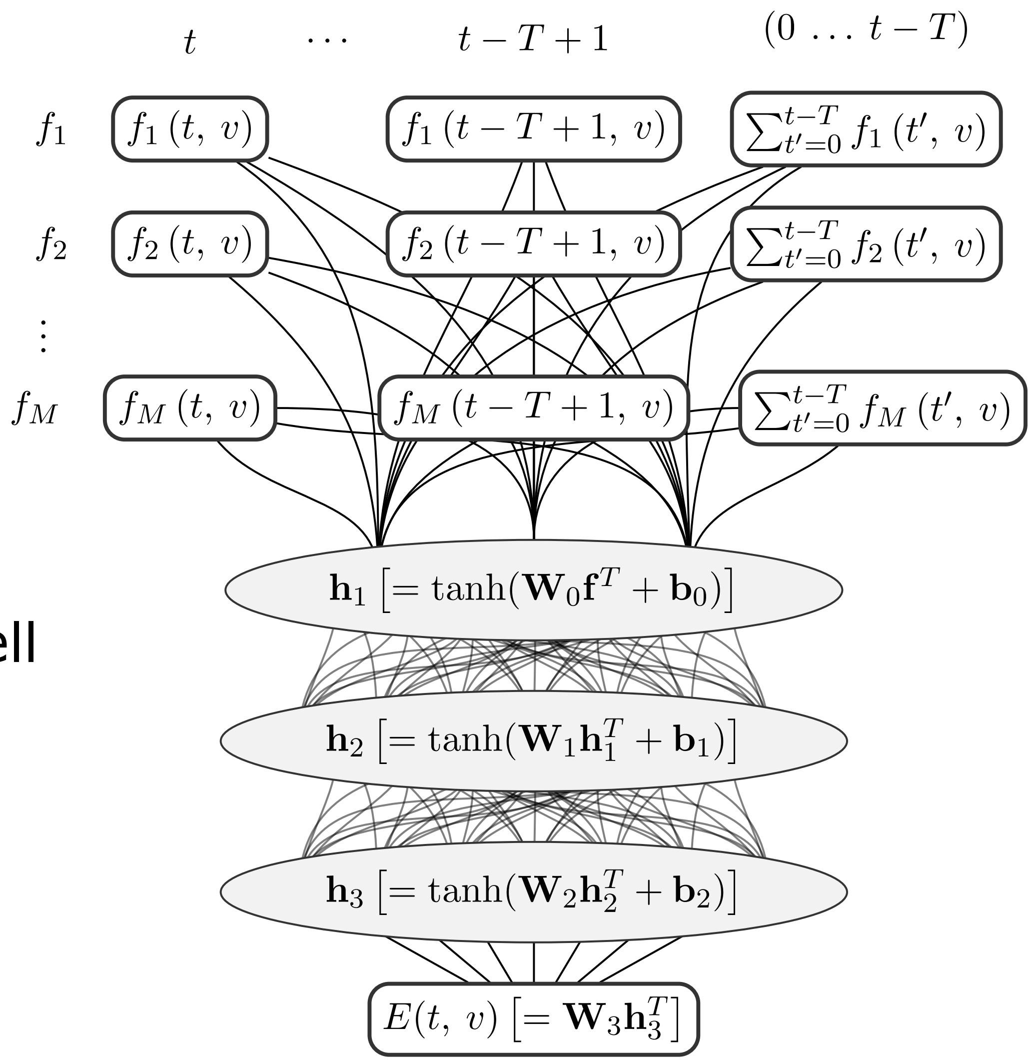


Discriminative state tracking

- One example:
  Deep Neural Network

- Take features from
  previous 9 turns
  + summary of previous

- Designed to generalize well
  to other domains



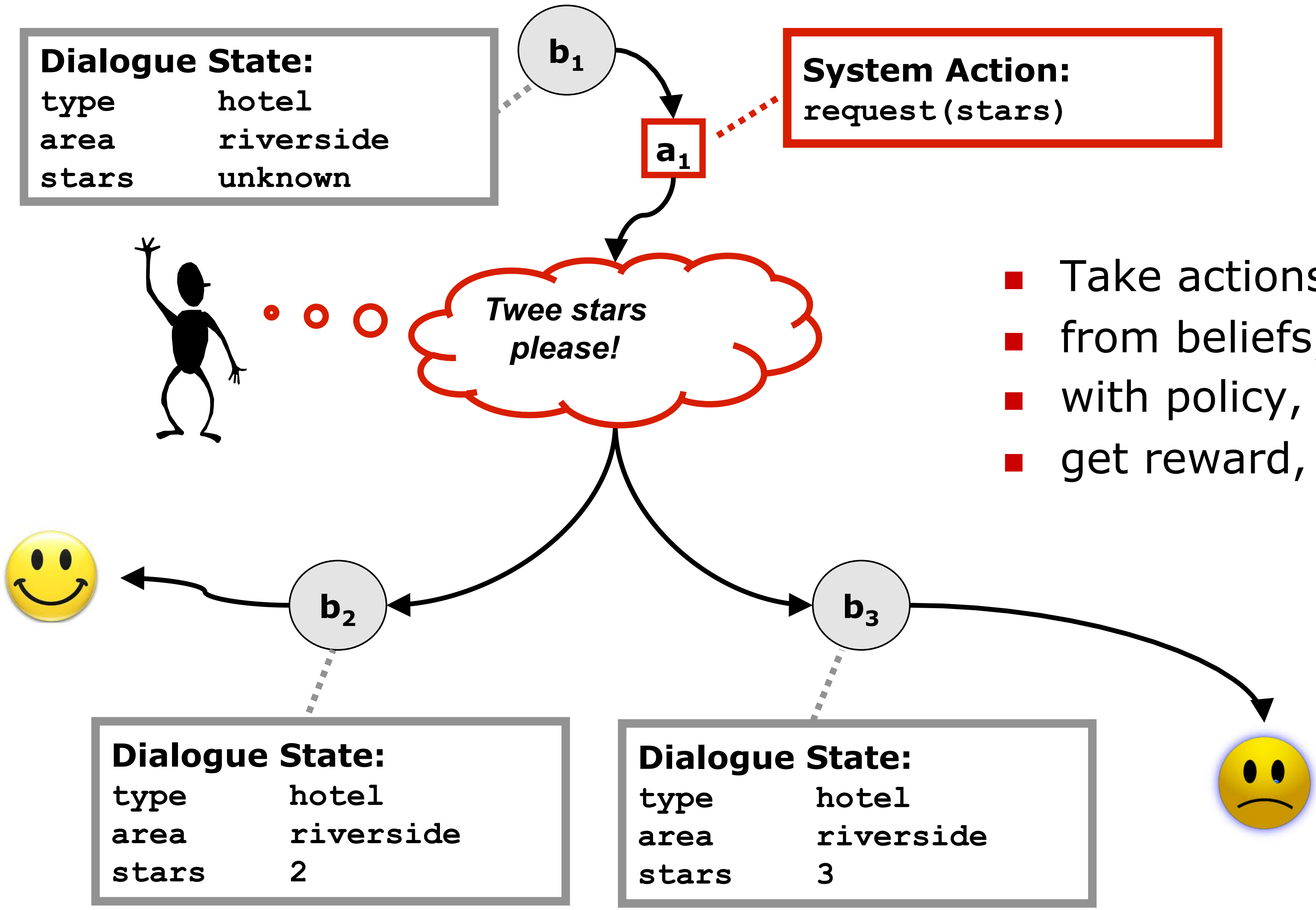
Henderson et al. SigDial 2013

Discriminative vs generative – summary

	Gen	Disc
Easily add complex dependencies		✓
Current best performance		✓
Generates for simulation	✓	
Trains from annotations	✓	✓
Currently trains unsupervised	✓	

Reinforcement learning – an example

Reinforcement learning – the idea



- Take actions, a
- from beliefs, b
- with policy, π
- get reward, r

Reinforcement learning – the aim

- Care about Expected Future Reward, or Q function:

$$Q^{\pi}(b,a) = E^{\pi}\left(\sum_t r(b_t, a_t)\right)$$

Expected Future Reward

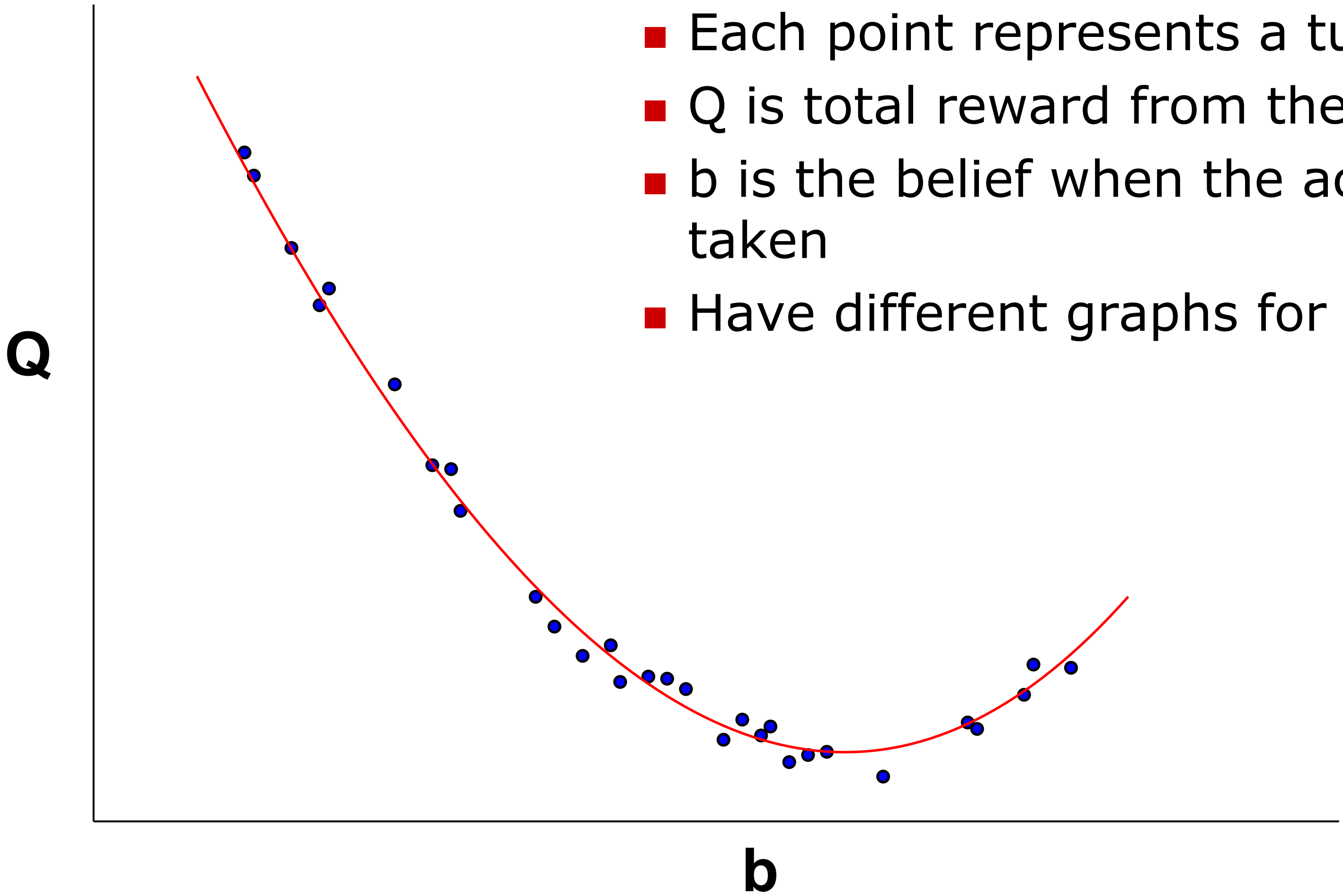
Reward at each time point

Choose π to maximise $Q^{\pi}(b_0, \pi(b_0))$:

Start State

Reinforcement learning – in practice

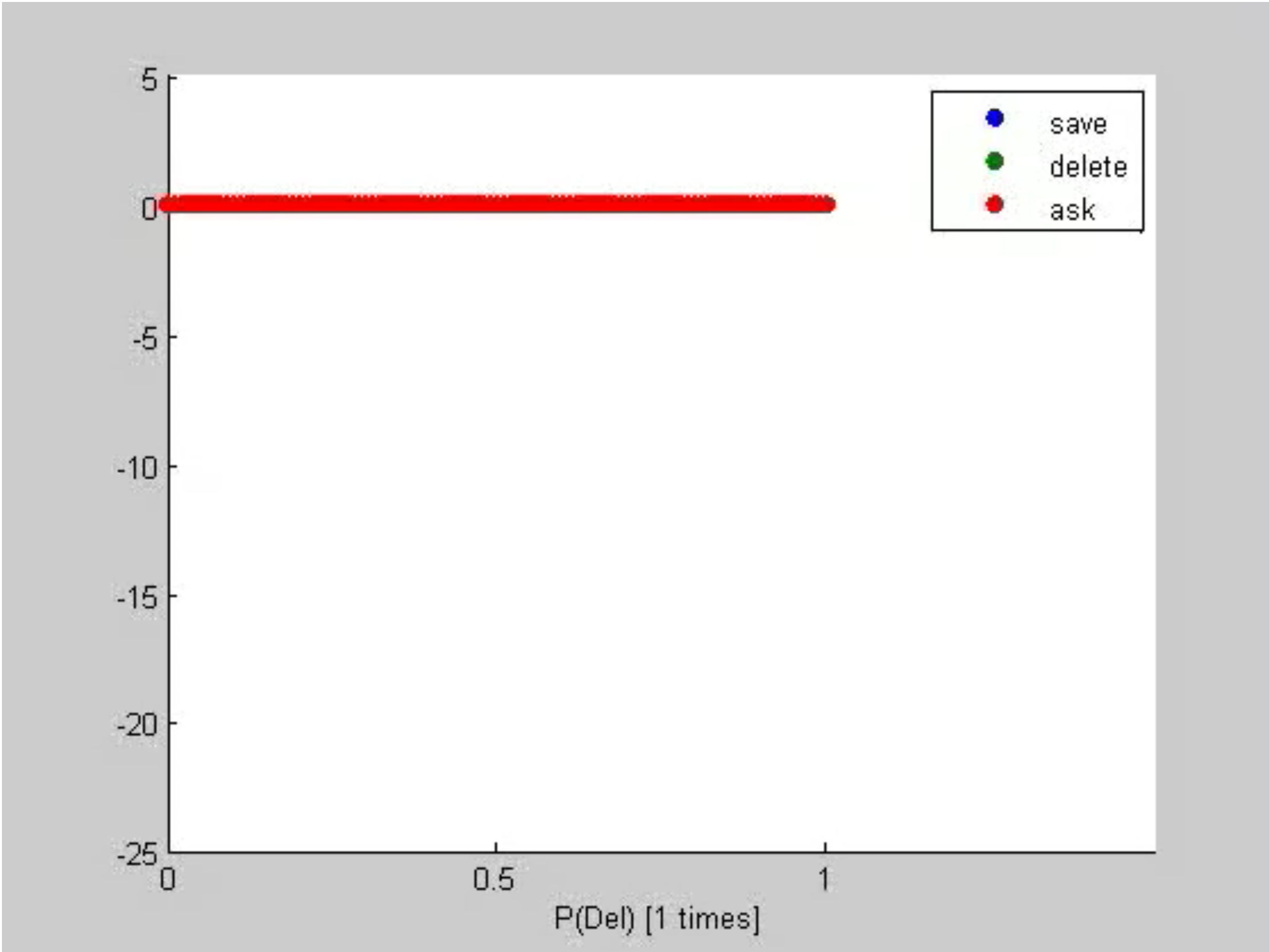
- In practice, we sample and make approximations



- Each point represents a turn
- Q is total reward from then on
- b is the belief when the action was taken
- Have different graphs for each a

Reinforcement learning – An example

Voicemail – Save message, Delete message or Ask user?

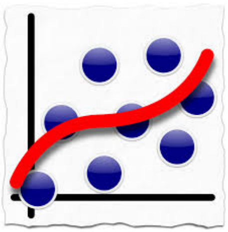


Reinforcement learning – the rewards

Where do we get the rewards?

1. Ask

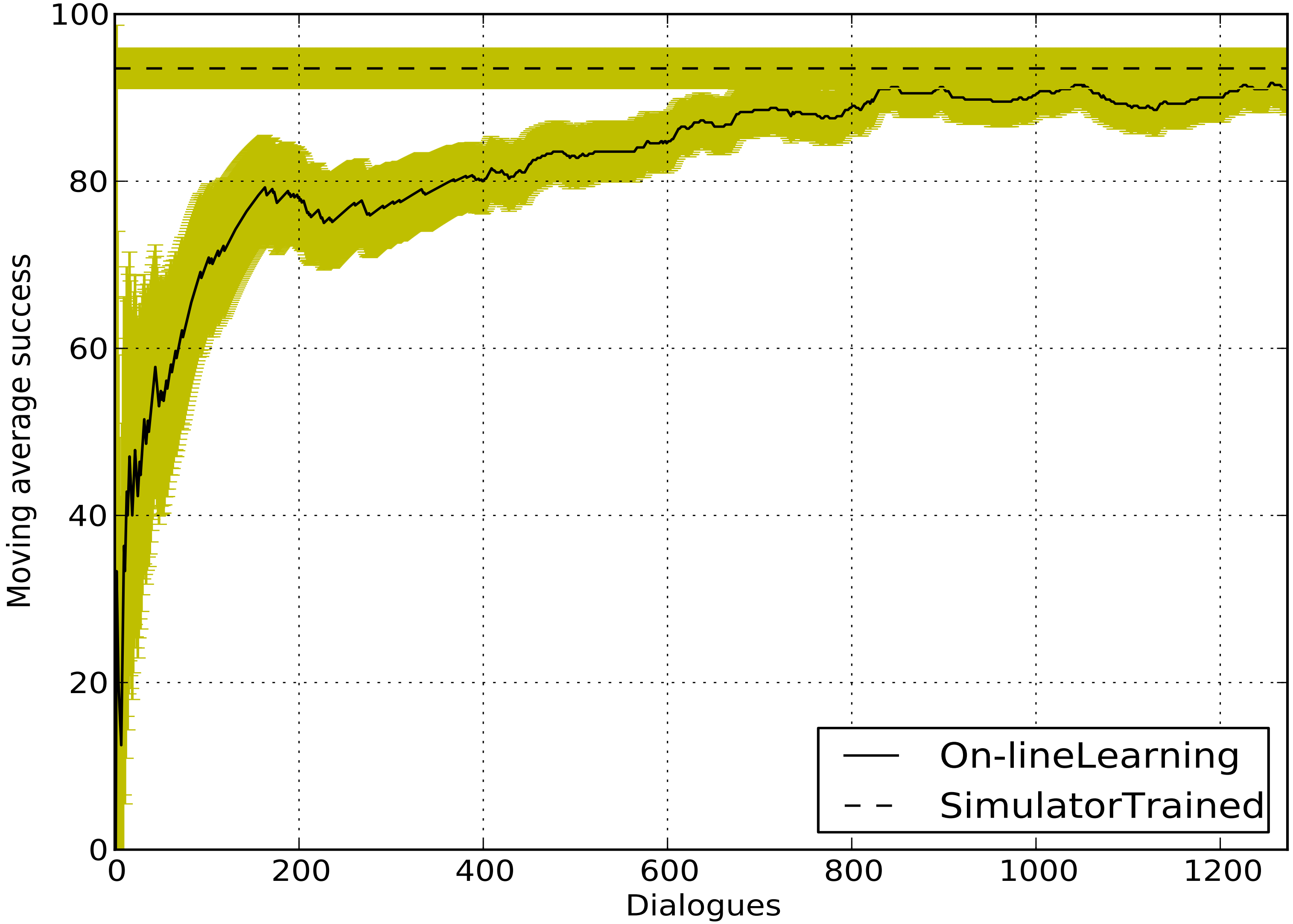
2. Simulate



3. Estimate

4. Ask + Estimate + Screen

Reinforcement learning – with humans

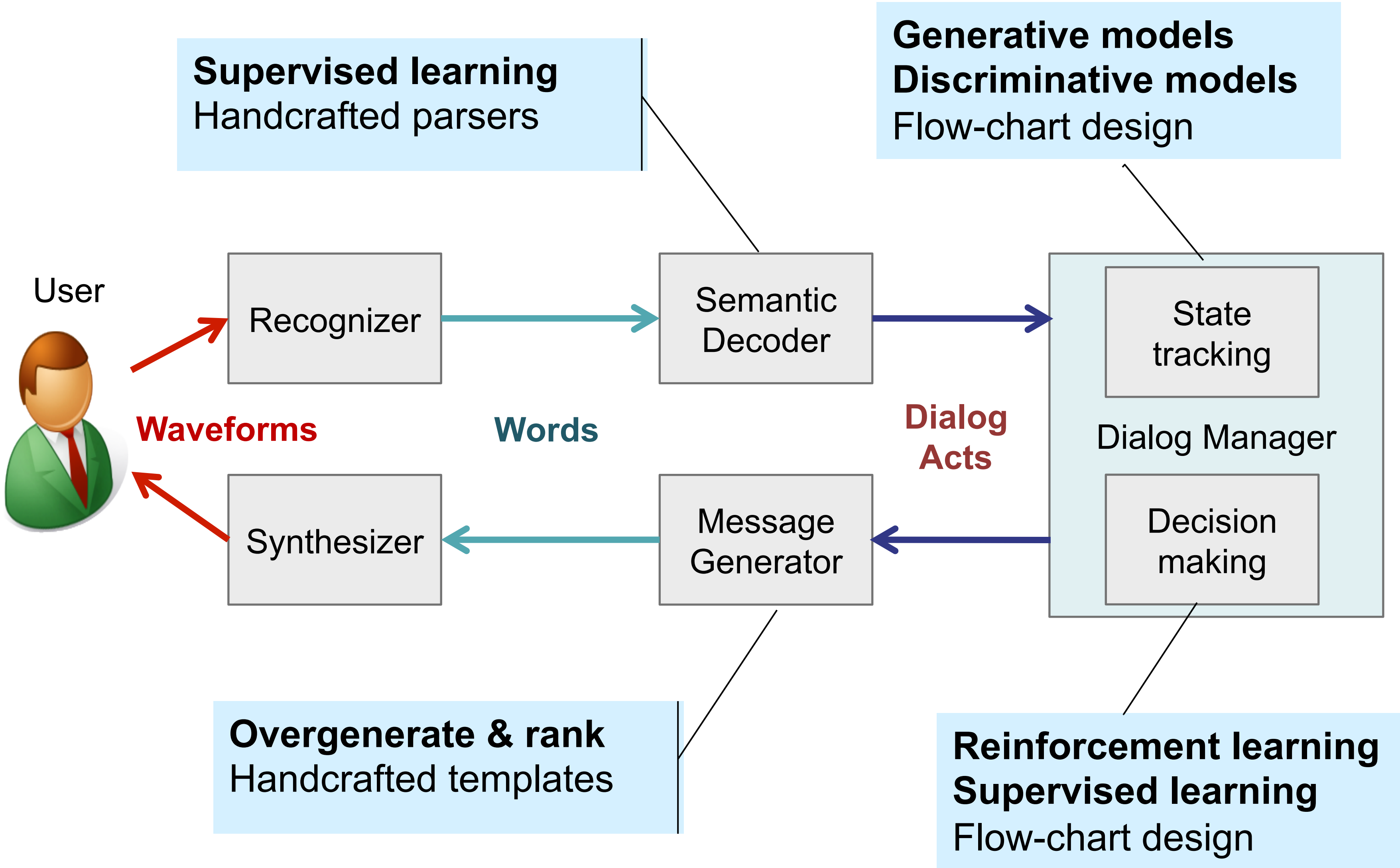


Reinforcement learning – with humans

	Simulator trained	On-line trained
Evaluation dialogues	400	410
Reward	11.6 +/- 0.4	13.4 +/- 0.3
Success (%)	93.5 +/- 1.2	96.8 +/- 0.9

Gasic et al. ICASSP 2013

Spoken dialogue – current techniques



Turn taking

Actually it’s really hard to know when to speak!

1. **Voice Activity Detectors**
2. **Continuous ASR**
3. **Reinforcement learn**

Tools – Corpora

- ATIS
  - Flight timetables
  - 5000 utterances

- Lets Go – DSTC 1
  - Bus timetable domain
  - 15 000 of live dialogs
  - Audio, Annotated text, Annotated fixed goals

- Cambridge Restaurants – DSTC 2
  - Restaurant information
  - 6 000 dialogs with mechanical turkers
  - Audio, Annotated text, Annotated semantics, Annotated goals (which can change)
  - 2014 Challenge task

Tools – Software

- Olympus dialog system (Hand-crafted)
  - Dan Bohus - http://www.cs.cmu.edu/~dbohus/
  - Desktop-based system
  - C++
- webdialog framework:
  - Matt Henderson - bitbucket.org/matthen/webdialog
  - Chrome-based speech recognition
  - Python

Future directions

- Better algorithms:
  - Semantics
  - State tracking
  - Policy learning
  - Generation
- Changing domains
- Open domain systems
- Removing the fixed turn model (incremental dialog)

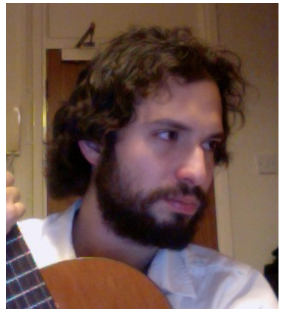
Thanks – Blaise Thomson

All joint work with Cambridge Dialogue Systems group

Pirros Tsiakoulis

Dongho Kim

Milica Gasic



Jorge Prombonas

Matt Henderson

Catherine Breslin

Martin Szummer

Steve Young