COM4511 Speech Technology: Spoken Dialogue Systems

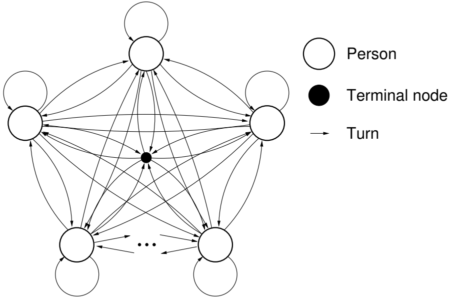
Anton Ragni

March 29, 2020

Why Dialogue Systems?

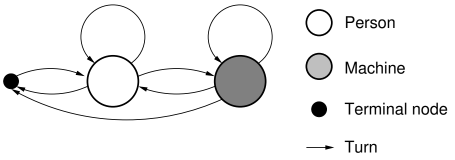
- One of the most popular applications of speech technology
  - also one of the most complex applications (later)
  - BUT progress so far has failed to meet market demand
- Integrates a large number of research areas
  - speech recognition (past lectures)
  - noise robustness/speech enhancement (past lecture)
  - natural language processing (COM4513)
  - speech synthesis (last lecture)
- The nature of the underlying phenomenon is highly complex
  - need adequate mathematical models and data
  - need large theoretical and engineering effort

Dialogue



- Written or spoken exchange between "people"
  - generally agreed upon definition
- BUT
  - number of people and turns may vary
  - turn taking not pre-determined
  - may target different objective and/or subjective goals
- The nature of dialogue is complex
  - no mathematical model proposed to date

Simple Dialogue

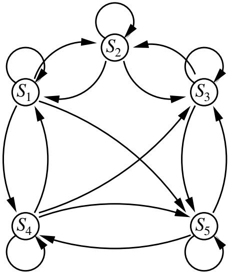


- A restricted form of a dialogue
  - two parties (human and machine)
  - multi-turn exchange
  - turn taking initiative on either side
  - turn interruption typically on the human side
  - objective (information gain) and/or subjective (likeness) goal
- Partially adequate mathematical model available
  - partially observable Markov decision process

Anton Ragni

Markov Process (Revisited)

(a) Dynamic Bayesian Network



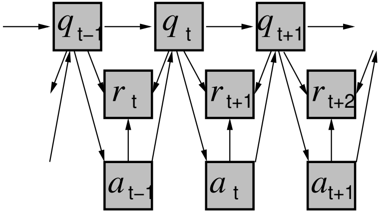
(b) Example Topology

- Key elements
  - states: $S_1, \ldots, S_N$
  - initial state distribution: $\boldsymbol{\pi} = \{P(S_i)\}_{i=1 \ldots N}$
  - transition probabilities: $\boldsymbol{\Pi} = \{P(S_i|S_j)\}_{i=1 \ldots N, j=1 \ldots N}$
- Probability of state sequence

$$P(\boldsymbol{q}_{1:T}) \approx \prod_{t=1}^{T} P(q_t|q_{t-1})$$

Markov Decision Process

(a) Markov Process (MP)



(b) Markov Decision Process (MDP)

- A version of Markov process
  - takes actions prior to making transitions
  - receives rewards for taking actions
- Reduces to Markov process if
  - only one action available per state
  - all rewards are the same
- Popular model for describing interactions with environments

Value Function

- Objective is to find policy $\boldsymbol{a}^*$ (actions) maximising expected discounted reward

$$V^{\boldsymbol{a}^*}(q_1 = S_i) = \mathcal{E}^{\boldsymbol{a}^*}_{\substack{a_t \sim P(a|q_t) \\ q_t \sim P(q|q_{t-1}, a_{t-1})}} \left\{ \sum_{t=1}^{T} \gamma^{t-1} r(q_t, a_t) \right\}$$

  - transition probabilities and rewards usually stationary
  - finite $T < \infty$ (FH) and infinite $T = \infty$ (IF) horizon versions
- Bellman equations provide framework for policy optimisation

$$V^{\boldsymbol{a}^*}(S_i) = \max_{a} \left\{ \underbrace{r(S_i, a) + \gamma \mathcal{E}\{V^{\boldsymbol{a}^*}(f(S_i, a))\}}_{Q(S_i, a)} \right\}$$

  - state update $S_j = f(S_i, a)$, quality function $Q(S_i, a)$
  - can be solved using dynamic programming (FH) or fixed-point (IH) methods

Anton Ragni

Example: Mini Tesco Bakery (Stochastic Inventory Model)

- Reward (expected revenue minus ordering and holding cost)

$$r(q_{t+1}, a_t, q_t) = F(q_t + a_t - q_{t+1}) - O(a_t) - H(q_t + a_t)$$

  - revenue $F(u) = 8u$
  - order cost $O(u) = K + o(u)$, where $K = 4$ is placement and $o(u) = 2u$ per-unit costs
  - holding cost $H(u) = u$
- Transition probabilities

$$P(q_{t+1}|q_t, a_t) = \begin{cases} p_{q_t+a_t-q_{t+1}}, & \text{if } M \geq q_t + a_t \geq q_{t+1} > 0 \\ q_{q_t+a_t}, & \text{if } M \geq q_t + a_t \text{ and } q_{t+1} = 0 \\ 0, & \text{otherwise} \end{cases}$$

  - partial $\boldsymbol{p} = [\frac{1}{4}\ \frac{1}{2}\ \frac{1}{4}\ 0]$ and full $q_i = \sum_{j=i}^{\infty} p_j$ demand probabilities
- Exercise:
  - compute all reward and transition probabilities
  - compute expected reward $r(q_t, a_t) = \sum_{q_{t+1}} P(q_{t+1}|q_t, a_t) r(q_{t+1}, q_t, a_t)$

Anton Ragni

Example (continued)

- Find optimal policy using Bellman recursion

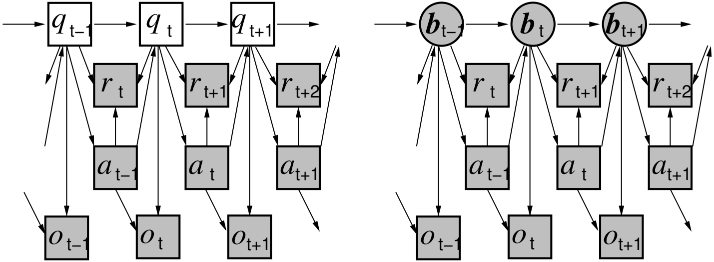
$$V_t^{\mathbf{a}^*}(q_t) = \max_{a_t} \Bigg\{ \underbrace{r(q_t, a_t) + \gamma \sum_{q_{t+1}} P(q_{t+1}|q_t, a_t) V_{t+1}^{\mathbf{a}^*}(q_{t+1})}_{Q_t(q_t, a_t)} \Bigg\}$$

  - value function at time t and state S_i if optimal policy is executed till time $T \leq \infty$
  - initialise $V_{T+1}^{\mathbf{a}^*}(S_i) = 0$ for all i
- Exercise:
  - show that value function and optimal actions are as follows

t	$V_t^{\mathbf{a}^*}(q_t)\|a_t^*$ $q_t = 0$	$q_t = 1$	$q_t = 2$	$q_t = 3$
4	$0\|-$	$0\|-$	$0\|-$	$0\|-$
3	$0\|0$	$5\|0$	$6\|0$	$5\|0$
2	$2\|2$	$\frac{25}{4}\|0$	$10\|0$	$\frac{21}{2}\|0$
1	$\frac{67}{16}\|3$	$\frac{129}{16}\|0$	$\frac{194}{16}\|0$	$\frac{227}{16}\|0$

  - discuss application of this (non-stationary) policy to bakery operation

Partially Observable Markov Decision Process



(a) Partially Observable MDP   Belief MDP

- Many environments cannot be observed directly
  - belief state expresses uncertainty in the environment state (sufficient statistics)

$$b_{t+1}(S_j) = P(S_j|\boldsymbol{o}_{1:t+1}, \boldsymbol{a}_{1:t}) = \frac{P(o_{t+1}|S_j, a_t)\sum_{i=1}^{N} P(S_j|S_i, a_t)b_t(S_i)}{\sum_{j=1}^{N} P(o_{t+1}|S_j, a_t)\sum_{i=1}^{N} P(S_j|S_i, a_t)b_t(S_i)}$$

  - belief state update (what are $\boldsymbol{P}^{\text{o}}_{o_{t+1}}$ and $\boldsymbol{P}^{\text{t}}_{a_t}$?)

$$\boldsymbol{b}_{t+1} = \boldsymbol{f}(\boldsymbol{b}_t, o_{t+1}, a_t) = \frac{\boldsymbol{P}^{\text{o}}_{o_{t+1}}\boldsymbol{P}^{\text{t}}_{a_t}\boldsymbol{b}_t}{\mathbf{1}^{\mathsf{T}}\boldsymbol{P}^{\text{o}}_{o_{t+1}}\boldsymbol{P}^{\text{t}}_{a_t}\boldsymbol{b}_t}$$

Anton Ragni

Example: Mini Coffee Shop (Machine Replacement Problem)

- Transition probabilities

$$\boldsymbol{P}_0^{\mathrm{t}} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} = \{P(q_{t+1}|q_t, a_t = 0)\}, \quad \boldsymbol{P}_1^{\mathrm{t}} = \begin{bmatrix} 1 & 0 \\ \theta^{\mathrm{d}} & 1 - \theta^{\mathrm{d}} \end{bmatrix} = \{P(q_{t+1}|q_t, a_t = 1)\}$$

  - states {poor=0, good=1}, actions {replace=0, use=1}
  - "poor" state cannot be escaped other than through replacement
  - probability to transition into "poor" state is θ^{d}
- Observation probabilities

$$\boldsymbol{P}_0^{\mathrm{o}} = \begin{bmatrix} \theta^{\mathrm{p}} & 0 \\ 0 & 1 - \theta^{\mathrm{g}} \end{bmatrix}, \quad \boldsymbol{P}_1^{\mathrm{o}} = \begin{bmatrix} 1 - \theta^{\mathrm{p}} & 0 \\ 0 & \theta^{\mathrm{g}} \end{bmatrix}$$

  - observations {poor=0, good=1}, poor θ^{p} and qood θ^{g} quality coffee self-probabilities
- Immediate rewards or negative costs
  - replacement $r(q_t, a_t = 0)$ and operation $r(q_t, a_t = 1)$ rewards
  - belief state reward

$$r(\boldsymbol{b}_t, a_t) = \sum_{i=1}^{N} b_t(S_i) r(S_i, a_t)$$

Anton Ragni

Example (continued)

- Bellman recursion for finding optimal policy

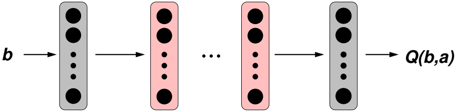
$$V_t^{\boldsymbol{a}^*}(\boldsymbol{b}_t) = \max_{a_t} \left\{ \underbrace{r(\boldsymbol{b}_t, a_t) + \gamma \sum_{o_{t+1}} P(o_{t+1}|\boldsymbol{b}_t, a_t) V_{t+1}^{\boldsymbol{a}^*}(\boldsymbol{f}(\boldsymbol{b}_t, o_{t+1}, a_t))}_{Q_t(\boldsymbol{b}_t, a_t)} \right\}$$

  - non-trivial to solve due to continuous belief state space
  - alternative options: discretise, approximate
- Exercise:
  - solve POMDP by discretising $\lambda \in [0, 1]$ parameterising belief state $\boldsymbol{b}_t = \begin{bmatrix} 1 - \lambda_t \\ \lambda_t \end{bmatrix}$

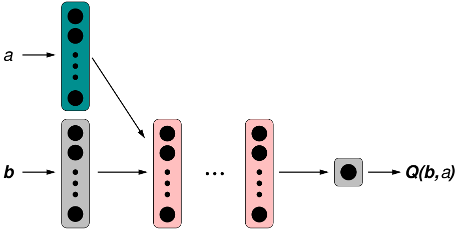
Reinforcement Learning

- Many environments can be challenging to model
  - large state and action spaces
  - lack of analytical forms
  - simulation-only environments
  - trial-and-error setting
- Example:
  - how many transition probabilities needed when $|\mathcal{S}| = 10,000$ and $|\mathcal{A}| = 1,000$?
  - how would you estimate transition probabilities in game playing (chess, Go)?
- Reinforcement learning offers practical solutions for challenging environments
  - learn optimal policy by interacting with environment
  - model-free and model-based environments possible
- Standard reinforcement learning approaches
  - Q-learning: estimate $Q(\boldsymbol{b}, a)$ using any machine learning approach
  - policy search: estimate parametric policy model bypassing value/quality functions
  - actor-critic methods: hybrid of the above

"Deep" Q-Learning



(a) One-to-many



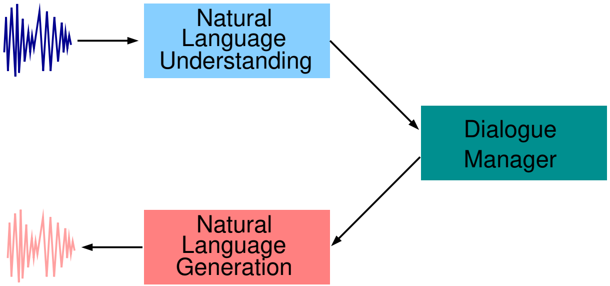
(b) One-to-one

- Use stochastic optimisation to learn Q-function (several architectures possible)

$$\mathcal{L}(\boldsymbol{\theta}^{(i)}; \mathcal{D}) = \mathcal{E}_{(\boldsymbol{b},a,\boldsymbol{b}')\sim\mathcal{D}} \left\{ \left(Q(\boldsymbol{b}', a; \hat{\boldsymbol{\theta}}) - Q(\boldsymbol{b}, a; \boldsymbol{\theta}^{(i)}) \right)^2 \right\}$$

  - sample of experiences $\mathcal{D}$, target $\hat{\boldsymbol{\theta}}$ and current $\boldsymbol{\theta}^{(i)}$ network
  - targets $Q(\boldsymbol{b}', a; \hat{\boldsymbol{\theta}})$ change with parameter changes (unlike supervised learning)

Spoken Dialogue Systems



- Key components
  - natural language understanding: convert message into observation o_t
  - dialogue manager: update belief state $\boldsymbol{b}_t$ and chose action a_t based on $Q(\boldsymbol{b}_t, a_t)$
  - natural language generation: convert action into message and estimate reward
- Alternative architectures
  - rule-based or scripted
  - information retrieval

Speech recognition and synthesis are key parts of natural language processing modules

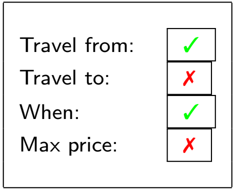
Natural Language Understanding

A life-time?	All-time NBA player	She is 88-years old!	Sounds fun, right?	I can't believe that!
▶ clarification	▶ Christian Wood	▶ who?	▶ Politics	▶ positive/negative
▶ info gathering	▶ Kareem Abdul-Jabbar		▶ Economy	▶ neutral
(a) Intent	(b) Named Entity	(c) Anaphora	(d) Topic	(e) Sentiment

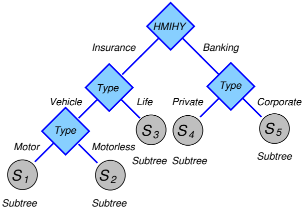
- Extract all necessary information from spoken content
  - BUT message uncertainty, semantic ambiguity
- Perfect speech transcription not possible
  - mistakes may cause harmful/harmless semantic changes
  - error mitigation is critical (confidence scores, embeddings)
- No known adequate semantic model
  - multiple semantic categories (above) but non-trivial to manually label
  - many conversational phenomena are not accounted for

Anton Ragni

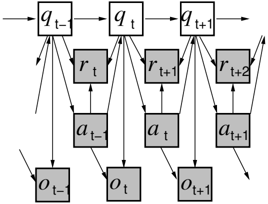
Dialogue Manager



(a) Slot-filling



(b) Flow-chart



(c) Statistical

- Certain tasks can be reduced to filling a pre-determined number of slots
  - most simple and widely deployed type
  - BUT unnatural and limited in scope
- Hand craft extensive flow-chart to describe a typical dialogue
  - trees can be arbitrary deep, cycles enable to create complex dialogues
  - BUT tedious and error-prone process, lack of generalisation
- Statistical methods enable dialogue model to be learnt from the data
  - enable unstructured and structured states
  - estimate and update "belief" over states by taking actions and receiving rewards
  - BUT lack of training data and limited generalisation

Very active area of research with Amazon hosting yearly Alexa prize challenges

Anton Ragni

Natural Language Generation

- Common expressions can be generated based on templates
  - example: `the train from ???  will arrive at platform ???`
  - high control ability, high production cost, limited scope
- Cast dialogue as an information retrieval problem
  - queries: user requests, hits: text excerpts from a large data set
  - system responses are spoken summaries of hits
- Use conditional neural language models $P(\boldsymbol{w}_{1:L}|\boldsymbol{c})$ to generate text
  - conditioning vector $\boldsymbol{c}$ encodes user requests and supplementary information
  - need to know how to encode diverse knowledge sources (Wiki, knowledge graphs)
- "Natural" natural language generation is a challenging problem†

Q: Who was Jason Mraz engaged to?	A: Jason Mraz was engaged to Tristan Prettyman
Q: When did WW2 end?	A: WW2 ended in 1945
Q: How many people live in the US?	A: There are over 300 million people living in the US

  - "natural" written text answers may not sound natural as spoken answers

† Examples from 17-billion-parameter language model by Microsoft

Evaluation

- No standardised evaluation measures exist, instead proxy measures are popular
  - success rate: subjective assessment by users (Likert scale, etc)
  - dialogue length: typically inversely correlated with above (exceptions: social setting, ...)
- Paradigm for dialogue system evaluation (PARADISE) most popular scheme
  - weighted combination of proxy measures, where weights estimated on labelled data
  - BUT mostly correlated with success rate and dialogue length (simple to optimised)
- Evaluation requires interactions with users
  - real: field deployment, need to have a reasonable baseline
  - paid: difficult to recruit large (statistical significance) number of subjects
  - simulated: wide coverage of dialogue space, hard to model real users

"Smart" Speakers

- Wide interest in conversational dialogue systems since Amazon Alexa deployment
  - numerous "smart" speakers are currently available

Language	Speakers (B)	Local Manufacturers
Mandarin	1.1	Alibaba, Baidu, Xiaomi
English	1.0	Apple, Amazon, Google
Hindustani	0.5	—
Spanish	0.5	—
Arabic	0.4	—
Malay	0.3	—
Russian	0.3	Yandex
Bengali	0.3	—

Account for slightly more than half of worlds population

- Goal-oriented dialogue systems with limited conversational ability
  - modular design (skills) enables to produce new extensions
  - BUT all extensions have limited scope and independent of each other

Summary

- This lecture explored spoken dialogue systems
  - simple dialogue model
  - Markov decision processes
  - reinforcement learning
- Focus on partially observable Markov decision processes
  - unobserved Markov process capable of taking actions and receiving rewards
  - Bellman equations and dynamic programming
- Discussed standard dialogue system architecture
  - natural language understanding
  - natural language generation
  - dialogue manager