A Survey of Task-oriented Dialogue Systems

HKUST CSE PhD Qualifying Examination

Kaixiang Mo

Committee Members:

Prof. Qiang Yang (Supervisor),

Dr. Yangqiu Song (Chairperson),

Dr. Xiaojuan Ma,

Prof. Pascale Fung (ECE)

16th January, 2017

Introduction & Motivation

- Dialogue systems have 2 major categories
  - Open-domain dialogue systems are used for daily chatting, and it **does not require feedback** on dialogue quality.
  - Task-oriented dialogue systems guide a user to finish certain task via dialogues, and the training of which **requires user feedback** on the dialogue quality.
- Task-oriented dialogue requires direct user feedback, while chatting based system does not require user feedback.
  - Is the target task completed?
  - How good is the dialogue quality?

	Open-domain Dialogue (X=customer, Y=system)
X_1	nothin much, and how's the book?!
Y_1	its good but i'm only like halfway through cuz i don't feel like reading. i'm so bored ...
X_2	that's good! i have the book but i'm bored too.

	Task-oriented Coffee Shopping Dialogue (X=customer, Y=system, R=feedback)
X_1	I would like a cup of coffee.
Y_1	What coffee would you like?
X_2	What coffee do you serve?
Y_2	We serve Espresso, Americano, Latte and Mocha.
X_3	I would like a cup of Latte.
Y_3	Hot Latte or Iced Latte?
X_4	Hot Latte.
R	**<Task-Completion-Feedback>**

Problem Definition

- User X and system Y have a dialog
  - User X's n-th utterance is $\boldsymbol{X}_n$
  - System Y's n-th utterance is $\boldsymbol{Y}_n$
- Input:
  - Current user utterance $\boldsymbol{X}_n$
  - Dialogue history $H_x = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_{n-1}\}, H_y = \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \dots, \boldsymbol{Y}_{n-1}\}$
- Output:
  - System Y's response sentence: $\boldsymbol{Y}_n$

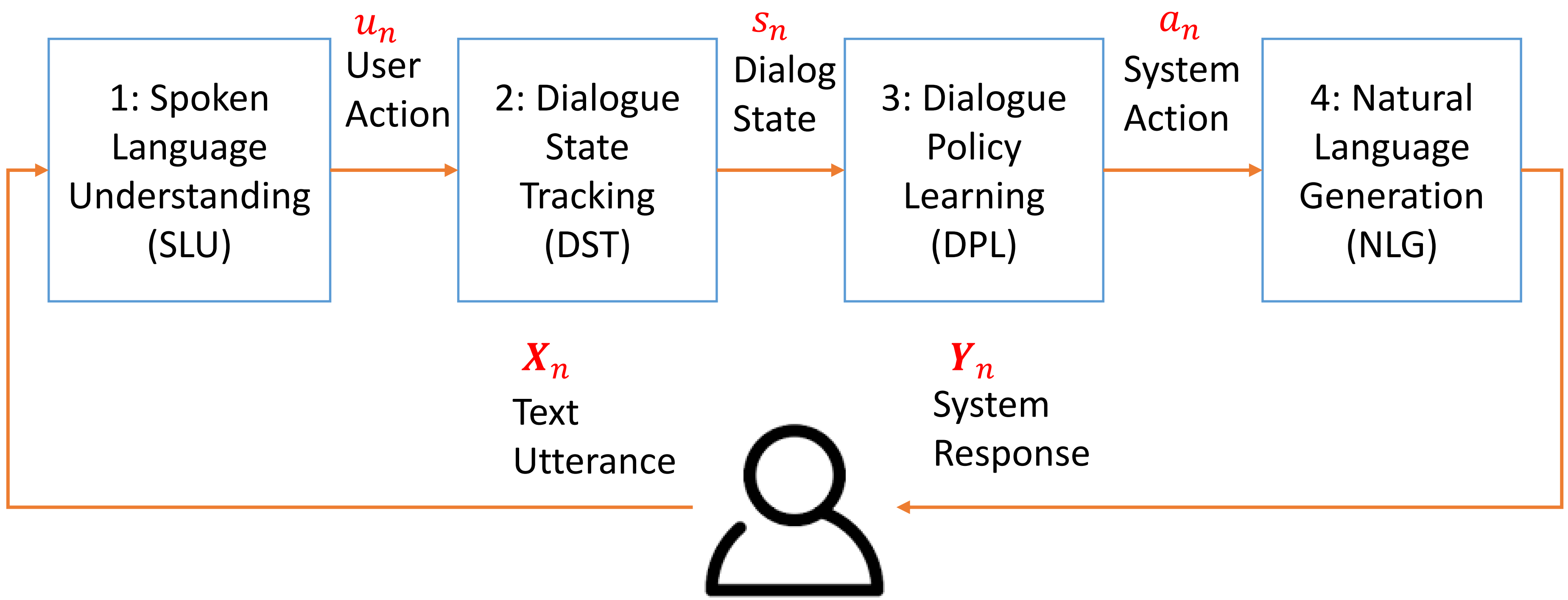
	Coffee Shopping Dialogue (X=customer, Y=system)
$\boldsymbol{X}_1$	I would like a cup of coffee.
$\boldsymbol{Y}_1$	What coffee would you like?
$\boldsymbol{X}_2$	What coffee do you serve?
$\boldsymbol{Y}_2$	We serve Espresso, Americano, Latte and Mocha.
$\boldsymbol{X}_3$	I would like a cup of Latte.
$\boldsymbol{Y}_3$	Hot Latte or Iced Latte?

Now

Dialogue System	
Input:	$\boldsymbol{X}_1, \boldsymbol{Y}_1, \boldsymbol{X}_2, \boldsymbol{Y}_2, \boldsymbol{X}_3$
Output:	$\boldsymbol{Y}_3$

Task-oriented Dialogue System Framework

- Task-oriented dialogue systems have 4 sub-tasks, each sub-task corresponds to a module in the system.



Sub-tasks in Task-oriented dialogue systems

- 1: Spoken Language Understanding (SLU)
  - SLU turns natural language into user intention and slot-values, and it takes $\boldsymbol{X}_n$ as input and outputs structured user action u_n
- 2: Dialogue State Tracking (DST)
  - DST tracks the current dialogue state, and it takes $\mathrm{s}_{\mathrm{n}-1}, \mathrm{a}_{\mathrm{n}-1}, u_n$ as input and outputs dialogue state s_n
- 3: Dialogue Policy Learning (DPL)
  - Policy decides which system action to take based on the dialogue state, and it takes dialogue state s_n as input and outputs system action a_n as output.
- 4: Natural Language Generation (NLG)
  - NLG turns a system action into natural language, and it takes the system action a_n as input and outputs the system response $\boldsymbol{Y}_n$

Tasks VS Techniques in Task-oriented Dialogue System (Detail)

Existing Work

Target: Personalization

		Modular Dialogue System					End-to-End Dialogue System	
		1:SLU	2:DST	3: Policy Learning (DPL)		4:NLG	General	Personalized
				General	Personalized			
None RL	None TL	CRF (Wang and Acero 2006; Raymond and Riccardi 2007) RNN (Yao *et al.* 2013; Mesnil *et al.* 2013, 2015; Liu and Lane 2015) LSTM (Yao *et al.* 2014)	HIS (Young *et al.* 2007) BUDS (Thomson *et al.* 2008) CRF (Lee 2013; Kim and Banchs 2014) RNN (Henderson *et al.* 2014c,d) LSTM (Zilka and Jurcicek 2015)			Conventional NLG (Walker *et al.* 2002; Stent *et al.* 2004) Corpus based (Oh and Rudnicky 2000; Mairesse and Young 2014; Angeli *et al.* 2010; Kondadadi *et al.* 2013) Neural network based (Mikolov *et al.* 2010; Wen *et al.* 2015a; Wen *et al.* 2015b)	RNN based (Wen *et al.* 2016b,a) Memory Network based (Bordes and Weston 2016)	
	TL	Instance based (Tur 2006) Model adaptation (Tür 2005) Parameter Transfer (Yazdani and Henderson)	Feature based transfer (Williams 2013; Ren et al. 2014) Model based transfer (Mrkši´c *et al.* 2015)			Model fine-tuning (Wen *et al.*, 2013) Curriculum learning Transfer (Shi *et al.* 2015) Instance Synthesis Transfer (Wen *et al.* 2016c)		
RL	None TL			Value-based RL (Williams 2008a,b; Young *et al.* 2010; Lefèvre *et al.* 2009; Li *et al.* 2009; Gaši´c and Young 2014; Gaši´c *et al.* 2013a; Daubigney *et al.* 2012b) Policy-based RL (Jurcícek *et al.* 2011; Su *et al.* 2016; Wen *et al.* 2016b) Actor-critic RL (Su et al. 2016; Jurcícek et al. 2011; Misu et al. 2012, 2010)			Policy network with Answer selection (Williams and Zweig 2016)	
	TL			Gaussian Process Transfer (Gasic *et al.* 2014; Gašic *et al.* 2013b; Gaši´c *et al.* 2015a) Bayesian Committee Machine Transfer (Gaši´c *et al.* 2015b)	Linear Model Transfer (Genevay and Laroche 2016). Gaussian Process Transfer (Casanueva et al. 2015)			

1:Spoken Language Understanding

- SLU takes $\boldsymbol{X}_n$ as input and outputs User Action $u_n = \{l_n, \boldsymbol{Z}_n\}$, where l_n is the intention class and $\boldsymbol{Z}_n$ are a sequence of slot-values.
  - Intention Classification
    - Classification Problem
    - $l_n = f(\boldsymbol{X}_n)$
    - Any classifiers can be used.
  - Slot-filling
    - Sequential Classification Problem
    - $\boldsymbol{Z}_n = f(\boldsymbol{X}_n)$, where $\boldsymbol{Z}_n = \{z_1, z_2, \dots\}$
- We survey methods on slot-filling problem because Intention Classification is relatively simple.

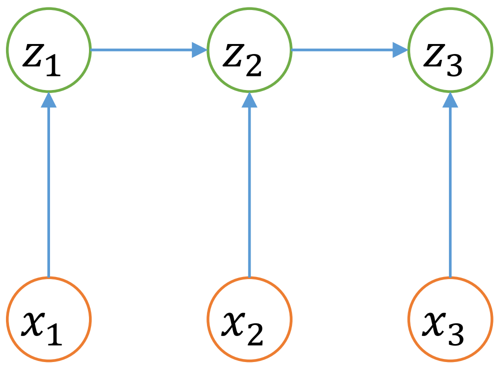
	Coffee Shopping Dialogue (X=customer, Y=system)
$\boldsymbol{X}_1$	I would like a cup of coffee.
$\boldsymbol{Y}_1$	What coffee would you like?
$\boldsymbol{X}_2$	What coffee do you serve?
$\boldsymbol{Y}_2$	We serve Espresso, Americano, Latte, Mocha, etc.
$\boldsymbol{X}_3$	I would like a cup of Latte.
$\boldsymbol{Y}_3$	Hot Latte or Iced Latte?
$\boldsymbol{X}_4$	Hot Latte.
$\boldsymbol{Y}_5$	What cup size do you want?
$\boldsymbol{X}_5$	Tall.

SLU	
Input:	$\boldsymbol{X}_3$="I would like a cup of Latte."
Output:	$\boldsymbol{u}_3$=$\{l_3, \boldsymbol{Z}_3\}$ l_3: Intention=Order, $\boldsymbol{Z}_3$: {CoffeeType=Latte}

1.1 Slot-filling Method: CRF (Lafferty *et al.* 2001)

- Wang and Acero (2006); Raymond and Riccardi (2007) use CRF for slot-filling. Xu and Sarikaya (2013) used CNN to extract feature for CRF.
- $p(z_t|x_t, z_{t-1}) = \frac{1}{Z}\exp(\boldsymbol{w}_{z_t}^T f(z_{t-1}, z_t, x_t) + \boldsymbol{b}_{z_t})$
- $f(z_{t-1}, z_t, x_t)$ is the feature vector including state transition probability.
- CRF can model label transition probability, but it consider fixed window size.

Slot-filling	
Input: $\boldsymbol{X}_n$	IPhone **7**. **7** IPhones.
Output: $\boldsymbol{Z}_n$	IPhone{Brand} 7{Generation}. 7{Quantity} IPhones{Brand}



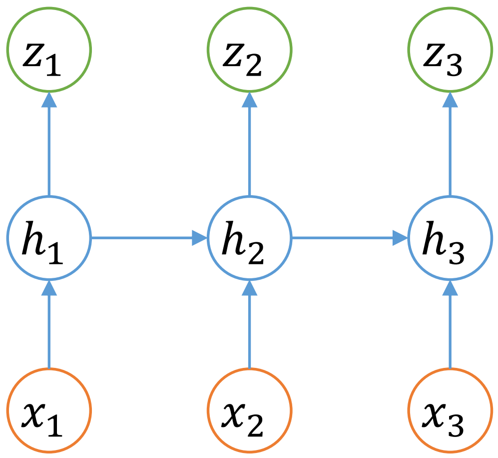
Pros:	**CRF can model label transition.**
Cons:	Considers only fixed window size.

1.2 Slot-filling Method: RNN (Goller and Kuchler 1996)

- Yao *et al.* (2013); Mesnil *et al.* (2013, 2015); Liu and Lane (2015) propose to use RNN for slot-filling problem.
- $\boldsymbol{h}_t = \sigma(\boldsymbol{W}_h \boldsymbol{h}_{t-1} + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b}_h)$
- $p(z_t | x_t, x_{<t}) = \frac{1}{Z}\sigma(\boldsymbol{w}_{z_t}^T \boldsymbol{h}_t + \boldsymbol{b}_{z_t})$
- Where $\boldsymbol{x}_t$ is the feature vector calculated in window centered at word x_t.

Slot-filling	
Input: $\boldsymbol{X}_n$	I want a new **phone** and I prefer **Apple**. The **fruit** looks good and I prefer **Apple**.
Output: $\boldsymbol{Z}_n$	…Apple{Company}. …Apple{Fruit}.

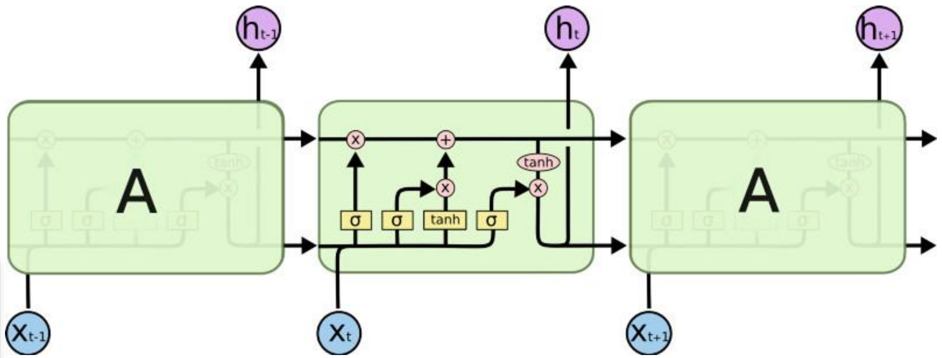
Pros:	**RNN can model label dependency of arbitrary length.**
Cons:	It is hard to train due to gradient vanishing problem.



1.3 Slot-filling Method: LSTM (Hochreiter and Schmidhuber 1997)

- Yao *et al.* (2014) propose to use LSTM (Hochreiter and Schmidhuber 1997) for slot-filling in the SLU.
- $\begin{bmatrix} \boldsymbol{i}_t \\ \boldsymbol{o}_t \\ \boldsymbol{f}_t \\ \hat{\boldsymbol{c}}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \boldsymbol{W} [\begin{smallmatrix} \boldsymbol{x}_t \\ \boldsymbol{h}_{t-1} \end{smallmatrix}]$
- $\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \hat{\boldsymbol{c}}_t$
- $\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t)$
- $z_t = \text{softmax}(\boldsymbol{h}_t)$
- It deal with gradient vanishing and gradient exploding problem in RNN, but LSTM requires more training data.

Slot-filling	
Input: $\boldsymbol{X}_n$	The **fruit** looks good, **but** I have to buy a new **phone** first. I prefer **Apple** IPhone.
Output: $\boldsymbol{Z}_n$	…Apple{Company} IPhone{Brand}.



Pros:	**LSTM can deal with gradient vanishing and gradient exploding problem.**
Cons:	LSTM requires more training data.

1.4 Evaluation of SLU

- The predicted slot values will be compared with ground truth slot-values labelled by human.
- SLU model comparison summary.

SLU Evaluation	
Metrics:	Intention Classification: Accuracy Slot-filling: F1 measure
Dataset:	Air Travel Information System (ATIS) (Dahl *et al.* 1994), Tourist Information (MEDIA) (Bonneau-Maynard *et al.* 2005), DARPA Communicator Travel Data (Walker *et al.* 2001)

	CRF	CRF-CNN	RNN	LSTM
F1 ATIS Mairesse *et al.* [2009]				
F1 ATIS Tür *et al.* [2013]	0.8373			
F1 ATIS Xu and Sarikaya [2013]	0.9100	0.9435	0.9411	
F1 ATIS Yao *et al.* [2013]	0.9109		0.9411	
F1 ATIS Mesnil *et al.* [2015]	0.9294		0.9498	
F1 ATIS Yao *et al.* [2014]	0.9294	0.9435	0.9411	0.9492

Transfer Learning for Dialogue System

- When the target domain does not have enough training data
  - Lacking <utterance, intention/slot-value> labelling data
  - Lacking <dialogue history, utterance, dialogue-state> labelling data
  - Lacking <dialogue state, system-action> labelling data
  - Lacking <system-action, response> labelling data
- Transfer learning can be used for
  - Building dialogue system for a new task/domain
  - Building personalized dialogue system for a targeted person

1.5 Transfer Learning for SLU

- When target domain do not have enough data, 3 kinds of transfer learning technique can be used:
  - Model adaptation for SLU (Tür 2005)
    - Regularize the target domain model with the KL divergence between source and target domain distribution.
  - Instance based transfer for SLU (Tur 2006)
    - Automatically map similar classes across domain, and transfer similar instances across domains.
  - Parameter transfer (Yazdani and Henderson 2015)
    - Use word embedding vector and parameter sharing between similar label classifiers, so similar classifiers have similar hyperplane.

Pros:	**Easy to use.**
Cons:	Source and target must use the same model configuration.

Pros:	**Works with any classifiers and do not require predefine instance similarity.**
Cons:	Time consuming.

Pros:	**Works in zeros-shot setting.**
Cons:	Requires extra data for learning word embedding.

2:Dialogue State Tracking

- DST takes s_{n-1}, a_{n-1}, u_n as input and outputs dialogue state s_n. s_n is the n-th dialogue state and a_n is the n-th system action.
  - State Representation
    - Dialogue state $s_n = \{g_n, u_n, h_n\}$
      - User Goal g_n
      - User Last Action u_n
      - Dialogue History $h_n = \{u_0, a_0, u_1, a_1, \dots, u_{n-1}, a_{n-1}\}$
  - State Tracking
    - $s_n = f(s_{n-1}, a_{n-1}, u_n)$
    - Can be modelled as a sequential classification problem

	Coffee Shopping Dialogue (X=customer, Y=system)
X_1	I would like a cup of coffee.
Y_1	What coffee would you like?
X_2	What coffee do you serve?
Y_2	We serve Espresso, Americano, Latte, Mocha, etc.
X_3	I would like a cup of Latte.

DST	
Input:	s_2={g_2={CoffeeType=?, Temp=?, Size=?}, u_2={Intention=Ask, {CoffeeType=?}}, h_2={u_1, a_1}}, a_2={Action=Inform,{CoffeeType=Espresso, Americano, Latte, Mocha}}, u_3={Intention=Order,{CoffeeType=Latte}}
Output:	s_3={g_3={CoffeeType=**Latte**, Temp=?, Size=?}, u_3={Intention=Order, {CoffeeType=Latte}} h_3={u_1, a_1, u_2, a_2}}

2.1 State Representation

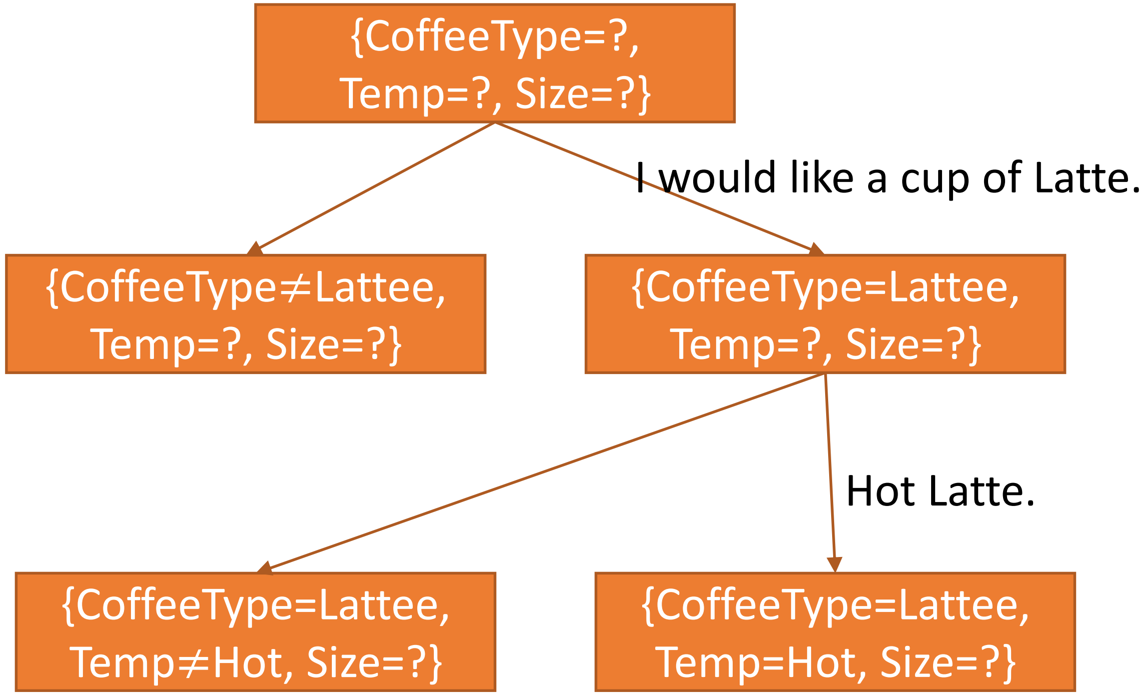
- Dialogue state $s_n = \{g_n, u_n, h_n\}$
  - User **Goal $\boldsymbol{g_n}$**
  - User Last Action u_n
  - Dialogue History $h_n = \{u_0, a_0, u_1, a_1, \ldots, u_{n-1}, a_{n-1}\}$
- The possible number of states is **exponential** to the number of slots in the domain, maintaining a distribution of all possible states requires a lot of resources.
- There are 2 ways to simplify state tracking
  - Hidden information state model
  - Bayesian model

User Goal	
CoffeeType:	N_1 possible values
Temp:	N_2 possible values
Size:	N_3 possible values
Possible Combination:	$N_1 * N_2 * N_3$ possible values

Image a toy domain with 5 slot, each slot has 10 possible value, the total number of states is 10^5.
Maintaining a transition probability matrix between all states requires $10^{10} * |a|$ space.

2.1.1 State Representation: Hidden Information State Model (HIS)

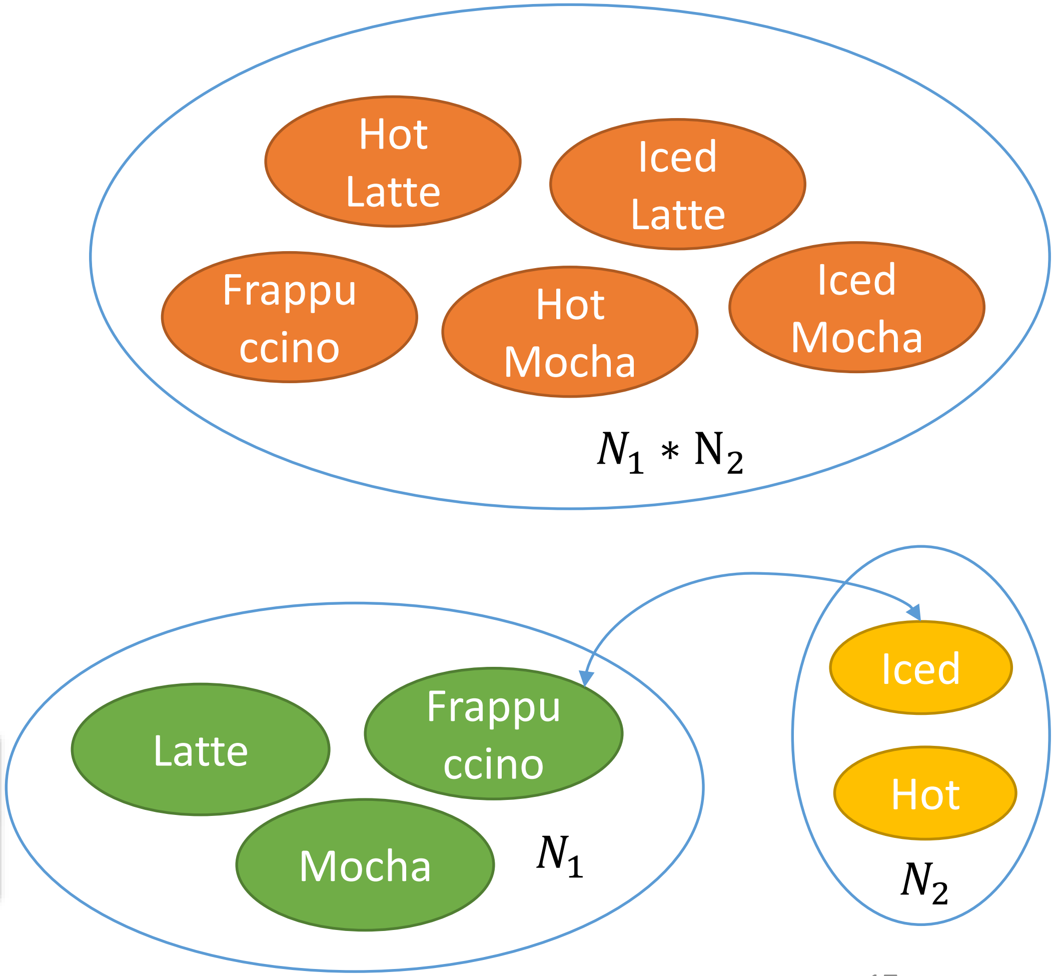
- Idea: use state grouping and state splitting to reduce tracking complexity.
- Young *et al.* (2007) propose HIS.
- All states are in the same group initially. {CoffeeType=?, Temp=?, Size=?}
- In each dialogue round, split one group into 2 partitions
  - {} is partitioned into {CoffeeType=Lattee} and {CoffeeType≠Lattee}
  - All states within the same partition has the same probability.



Pros:	**Can model arbitrary transition between any state pair.**
Cons:	However, number of groups to track is 2^n, so probability of only top N states are tracked.

2.1.2 State Representation: Bayesian Update of Dialogue States (BUDS)

- Idea: Assuming transition probability of different slots are independent of each other, or have very simple dependency.
- Reduce number of state from **exponential** to **linear**
- Thomson *et al.* (2008) propose to use dynamic bayesian network (*Murphy, Kevin 2002*) to track dialogue states.



Pros:	**Can track the probability of all possible states.**
Cons:	It cannot deal with complex transition.

2.2: State Tracking

- Static classifiers
  - $s_n = f(\{u_i, a_i\}_{i=1}^{n-1}, u_n)$
  - Linear Classifier, neural network and ranking model.
  - Features including size and confidence of each SLU output, probability of the top slot-value in each slot, etc.

Pros:	**Do not require s_{n-1}, work with any classifiers.**
Cons:	It did not consider state transition.

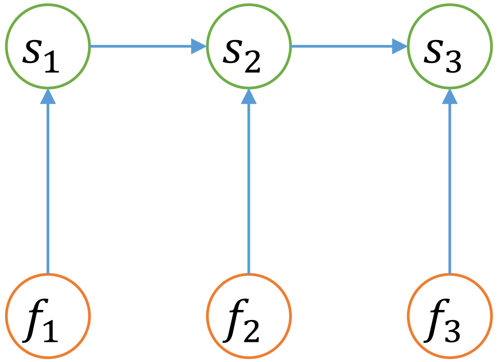
- Sequential classifiers
  - $s_n = f(s_{n-1}, a_{n-1}, u_n)$
  - CRF, RNN, LSTM

Pros:	**Consider state transition probability.**
Cons:	Error in s_{n-1} might affect s_n.

- We further introduce **sequential classifiers** because static classifiers are relative simple.

2.2.1 State Tracking: CRF (Lafferty *et al.* 2001)

- Lee (2013); Kim and Banchs (2014) use CRF for dialogue state tracking.
- $p(s_n | s_{n-1}, a_{n-1}, u_n) = \frac{1}{Z} \exp(\boldsymbol{w}_{s_n}^T f(s_{n-1}, s_n, a_{n-1}, u_n) + \boldsymbol{b}_{s_n})$
- $f(s_{n-1}, s_n, a_{n-1}, u_n)$ is the feature vector including state transition probability.



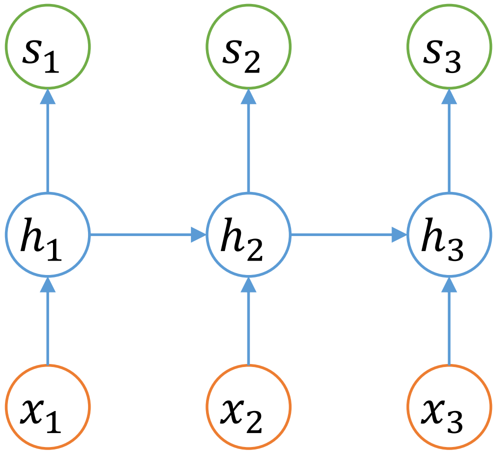
Pros:	**CRF can model state transition probability.**
Cons:	It consider fixed window size.

DST	
Input: $\boldsymbol{s}_1$, $\boldsymbol{a}_1$, $\boldsymbol{u}_2$	$\boldsymbol{s}_1$={Category=Phone}
Output: $\boldsymbol{s}_2$	$\boldsymbol{s}_2$={Category=Phone, Brand=Apple}

	Phone Shopping Dialogue (X=customer, Y=system)
$\boldsymbol{X}_1$	I would like a new phone.
$\boldsymbol{Y}_1$	Which brand do you prefer?
$\boldsymbol{X}_2$	Apple.

2.2.2 State Tracking: RNN (Goller and Kuchler 1996)

- Henderson *et al.* (2014c,d) use RNN for dialogue state tracking.
- $\boldsymbol{h}_n = \sigma(\boldsymbol{W}_h \boldsymbol{h}_{n-1} + \boldsymbol{U} f(a_{n-1}, u_n) + \boldsymbol{b}_h)$
- $p(s_n) = \frac{1}{Z}\sigma(\boldsymbol{w}_{s_n}^T \boldsymbol{h}_n + \boldsymbol{b}_{s_n})$
- Where $f(a_{n-1}, u_n)$ is the feature vector calculated at round n.



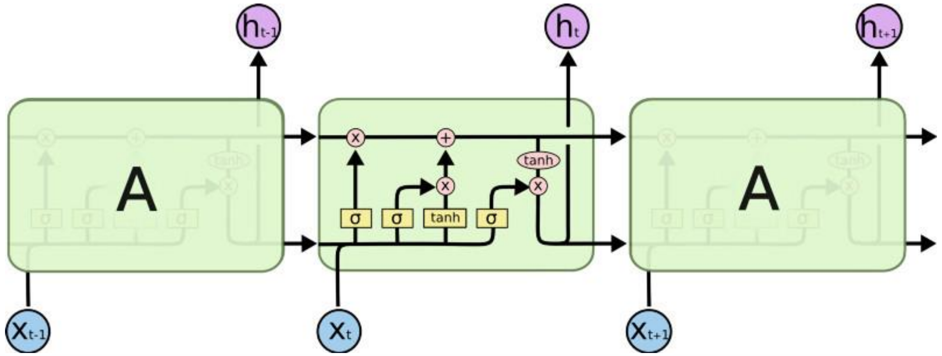
Pros:	**RNN can model label dependency of arbitrary length.**
Cons:	It is hard to train due to gradient vanishing problem.

DST	
Input: $\boldsymbol{s}_2$, $\boldsymbol{a}_2$, $\boldsymbol{u}_3$	$\boldsymbol{s}_2$={Category=Phone}
Output: $\boldsymbol{s}_3$	$\boldsymbol{s}_3$={Category=Phone, Brand=Apple}

	Phone Shopping Dialogue (X=customer, Y=system)
$\boldsymbol{X}_1$	I would like a new phone.
$\boldsymbol{Y}_1$	Which brand do you prefer?
$\boldsymbol{X}_2$	What is the available?
$\boldsymbol{Y}_2$	Samsung, Apple, HTC
$\boldsymbol{X}_3$	Apple.

2.2.3 State Tracking: LSTM (Hochreiter and Schmidhuber 1997)

- Zilka and Jurcicek (2015) use LSTM (Hochreiter and Schmidhuber 1997) for tracking dialogue states.
- $\begin{bmatrix} \boldsymbol{i}_n \\ \boldsymbol{o}_n \\ \boldsymbol{f}_n \\ \widehat{\boldsymbol{c}_n} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \boldsymbol{W} \begin{bmatrix} f(a_{n-1}, u_n) \\ \boldsymbol{h}_{n-1} \end{bmatrix}$
- $\boldsymbol{c}_n = \boldsymbol{f}_n \odot \boldsymbol{c}_{n-1} + \boldsymbol{i}_n \odot \widehat{\boldsymbol{c}_n}$
- $\boldsymbol{h}_n = \boldsymbol{o}_n \odot \tanh(\boldsymbol{c}_n)$
- $s_n = \mathrm{softmax}(\boldsymbol{h}_n)$



Pros:	**LSTM can deal with gradient vanishing and gradient exploding problem.**
Cons:	LSTM requires more training data.

DST	
Input: $\boldsymbol{s}_2$, $\boldsymbol{a}_2$, $\boldsymbol{u}_3$	$\boldsymbol{s}_2$={Category=Phone}
Output: $\boldsymbol{s}_3$	$\boldsymbol{s}_3$={Category=Phone, Brand=Apple}

	Phone Shopping Dialogue (X=customer, Y=system)
$\boldsymbol{X}_1$	The orange looks good.
$\boldsymbol{Y}_1$	Would you like some?
$\boldsymbol{X}_2$	I have to buy a new phone first.
$\boldsymbol{Y}_2$	Which brand do you prefer?
$\boldsymbol{X}_3$	Apple.

2.2.4 Evaluation of DST

- Predicted dialogue state will be compared with human labelled dialogue state.
- Dataset: DSTC1, DSTC2, DSTC3
- Tasks include
  - Goals: Desired Restaurant
  - Method: Search by constraint/name/alternative
  - Request: Phone number/Address
- DST model comparison summary.

	Utterance	Goals Food	Goals Area
S_1	Hello, How may I help you?		
U_1	I need a Persian restaurant in the south part of town.	Persian	South
S_2	What kind of food would you like?		
U_2	Persian.	Persian	South

DST Evaluation	
Metrics:	Top State Classification: Accuracy State probability tracking: L2
Dataset:	DSTC 1 (bus information), DSTC 2 (restaurant information) (Henderson *et al.* 2014a) DSTC 3 (coffee shop, pub and restaurant) (Henderson *et al.* 2014b).

		Goal		Method		Request	
Method	Dataset	Accuracy	L2	Accuracy	L2	Accuracy	L2
Linear CRF	DSTC2	0.601	0.648	0.904	0.155	0.960	0.073
RNN	DSTC2	0.768	0.346	0.940	0.095	0.978	0.035
LSTM	DSTC2	0.72	0.64	0.93	0.14	0.97	0.06
Ranking	DSTC2	0.78	0.35	0.95	0.08	0.98	0.04

2.2.5 Transfer learning for DST

- When target domain do not have enough data, 2 kinds of transfer learning technique can be used:
  - Feature based transfer for DST
    - Build general domain independent features, so that the trained models can be used in multi-domain setting.
    - Williams (2013) propose to use shared synthetic features.
    - Ren et al. (2014) propose to share the dialogue state tracking model across different domains by using a domain dependent feature set.
  - Model based transfer for DST
    - Adapts an general domain independent tracking model with the domain dependent datasets
    - Mrkšic *et al.* (2015) propose to initialize domain dependent tracker RNN with a general RNN.

Pros:	**Easy to implement.**
Cons:	Requires careful feature design and lots of human effort.

Restaurant: <Food> ~= Hotel: <Area>
RNN tile the weight of different slots.

Pros:	**It can transfer between fine granularity slots.**
Cons:	Only work on delexicalised data.

3:Dialogue Policy Learning

- Policy Learning takes dialogue state s_n as input and outputs system action a_n as output.
- Dialogue Policy Learning can be modelled as a Reinforcement learning (Sutton and Barto, 1998) problem
  - Dialogue states can be model by MDP $\{S, A, P, R, \gamma\}$
  - Cumulative Reward is $G_n = \sum_k \gamma^k r_{n+k}$
- The objective is to maximize Cumulative Reward
- There are 3 Dialogue Policy Learning frameworks
  - Value-based RL
  - Policy-based RL
  - Actor-critic RL

	Task-oriented Coffee Shopping Dialogue (X=customer, Y=system, R=feedback)
X_1	I would like a cup of coffee.
Y_1	What coffee would you like?
X_2	What coffee do you serve?
Y_2	We serve Espresso, Americano, Latte and Mocha.
X_3	I would like a cup of Latte.
Y_3	Hot Latte or Iced Latte?
X_4	Hot Latte.
R	<**Task-Completion-Feedback**>

Policy	
Input:	s_3={g_3={CoffeeType=Latte, Temp=?, Size=?}, u_3={Intention=Order, {CoffeeType=Latte}}, h_3={u_1, a_1, u_2, a_2}},
Output:	a_3={Action=Ask,{Temp=?}},

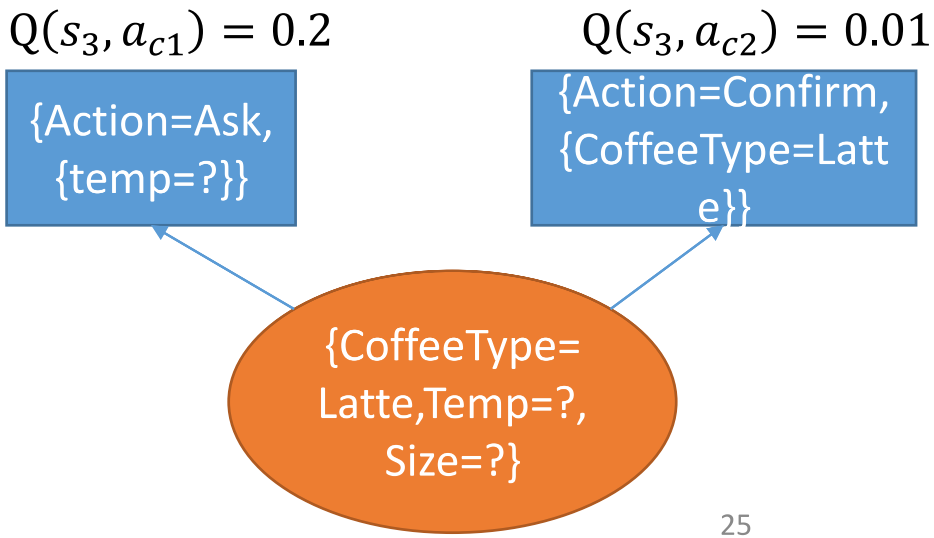
3.1 Policy Learning: Value-based RL

- Policy is modelled by Q-function, which predicts the cumulative reward starting from s_n taking action a_n and following policy π.
  - $Q(s_n, a_n) = E(\sum_{k=0}^{\infty} \gamma^k r_{n+k} \,|s_n, a_n)$
- The policy is determined by
  - $a_n = \mathrm{argmax}_{a'} Q(s_n, a')$
- Model can be trained with Q-learning (Watkins 1989)
  - $Q(s_n, a_n) = r_n + \max_{a'} \lambda Q(s_{n+1}, a') - Q(s_n, a_n)$

Pros:	**Easy to implement.**
Cons:	Cannot deal with large/continuous action space.

Now

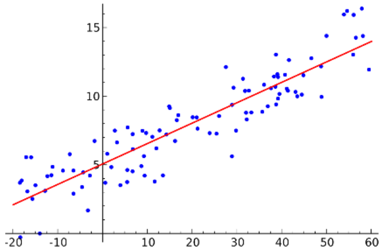
	Coffee Shopping Dialogue (X=customer, Y=system)
X_1	I would like a cup of coffee.
Y_1	What coffee would you like?
X_2	What coffee do you serve?
Y_2	We serve Espresso, Americano, Latte, Mocha, etc.
X_3	I would like a cup of Latte.
Y_3	Hot Latte or Iced Latte?
R	<**Task-Completion-Feedback**>

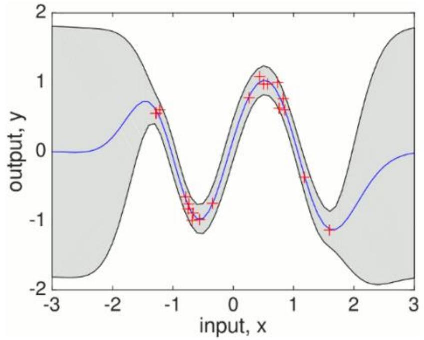


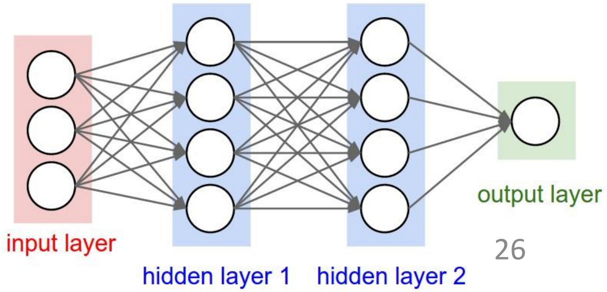
3.1.1 Value-based RL: Models for Q-function

	$\overline{a_1}$	$\overline{a_2}$
$\overline{s_1}$	0.6	0.2
$\overline{s_2}$	1.5	0.3

- Grid based Q-function (Williams 2008a,b; Young *et al.* 2010; Lefèvre *et al.* 2009)
  - Use discrete clusters to approximate states and actions.
  - Q-function is a lookup table $Q(s,a) = Table(\bar{s},\bar{a})$
- Linear model Q-function (Li *et al.* 2009)
  - Q-function is linear model $Q(s,a) = \sigma(\phi(s,a)^T W)$
- Gaussian Process based Q-function (Gaši´c and Young 2014; Gaši´c *et al.* 2013a)
  - Q-function is approximated by none-parametric Gaussian process
  - $Q^{\pi}(s,a) \sim GP(m(s,a), k((s,a),(s,a))$
- Neural Network based Q-function (Daubigney *et al.* 2012b)
  - Q-function is approximated by Neural Network.
  - $Q(s,a) = NN(\phi(s,a), W)$
  - $\phi_l(s,a) = \sigma(\phi_{l-1}(s,a)^T W)$
  - $\phi_l(s,a)$ is the neural network output of the l-th layer.







3.1.1 Value-based RL: Models for Q-function

- Grid based Q-function (Williams 2008a,b; Young *et al.* 2010; Lefèvre *et al.* 2009)
  - Use discrete clusters to approximate states and actions.
  - Q-function is a lookup table $Q(s,a) = Table(\bar{s},\bar{a})$
- Linear model Q-function (Li *et al.* 2009)
  - Q-function is linear model $Q(s,a) = \sigma(\phi(s,a)^T W)$
- Gaussian Process based Q-function (Gaši´c and Young 2014; Gaši´c *et al.* 2013a)
  - Q-function is approximated by none-parametric Gaussian process
  - $Q^\pi(s,a) \sim GP(m(s,a), k((s,a),(s,a))$
- Neural Network based Q-function (Daubigney *et al.* 2012b)
  - Q-function is approximated by Neural Network.
  - $Q(s,a) = NN(\phi(s,a), W)$
  - $\phi_l(s,a) = \sigma(\phi_{l-1}(s,a)^T W)$
  - $\phi_l(s,a)$ is the neural network output of the l-th layer.

Pros:	**Easy to implement.**
Cons:	Hard to decide number of clusters. Large quantization error.

Pros:	**Less quantization error and fast.**
Cons:	Require predefined features. Linear model has low representation capacity.

Pros:	**Can be trained with small sample.**
Cons:	Slow, high computational complexity and can only process small dataset.

Pros:	**Better representation ability.**
Cons:	Require a lot of data to train, overfitting easily on small dataset.

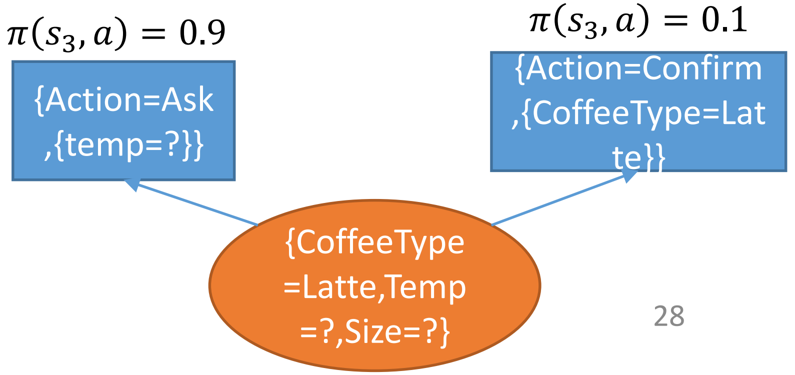
3.2 Policy Learning: Policy-based RL

- Policy is directly modelled by
  - $p(a_n|s_n) = \pi(s_n, a_n)$
  - $a_n \sim p(a_n|s_n)$
- Model can be trained with REINFORCE (Williams 1992) algorithm, which is to maximize the probability of the "good" episode.
  - $\theta = \theta + \alpha\Delta_\theta \log \pi_\theta(s_n, a_n)\boldsymbol{v}_n$
  - v_n is unbiased sample of the true future cumulative reward following π_θ
  - $v_n = \sum_k \gamma^k r_{n+k}$

	Coffee Shopping Dialogue (X=customer, Y=system)
$\boldsymbol{X}_1$	I would like a cup of coffee.
$\boldsymbol{Y}_1$	What coffee would you like?
$\boldsymbol{X}_2$	What coffee do you serve?
$\boldsymbol{Y}_2$	We serve Espresso, Americano, Latte, Mocha, etc.
$\boldsymbol{X}_3$	I would like a cup of Latte.
$\boldsymbol{Y}_3$	Hot Latte or Iced Latte?
$\boldsymbol{X}_4$	Hot Latte.
$\boldsymbol{Y}_4$	What cup size do you want?
$\boldsymbol{X}_5$	Tall.
R	**<Task-Completion-Feedback>**

Now

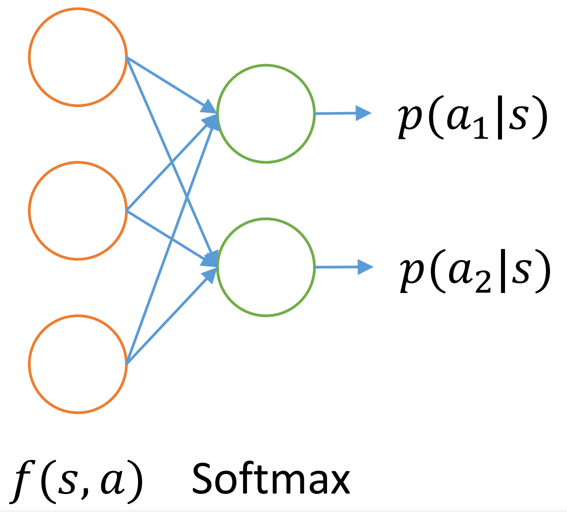
$$v_3 = \sum_k^N \gamma^k r_{3+k}$$



Pros:	**Policy-based RL can deal with continuous action space.**
Cons:	REINFORCE algorithm is unstable because v_n has large variance.

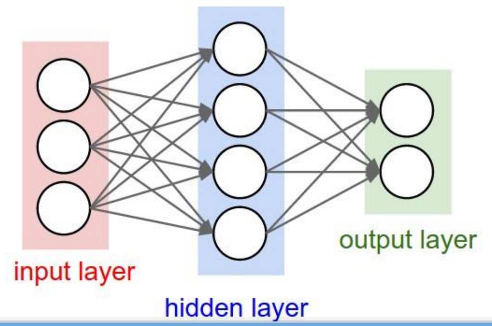
3.2.1 Policy-based RL: Modes for Policy function

- Softmax policy function (Jurcícek *et al.* 2011)
  - Action distribution is modelled by softmax
  - $p(a|s) = \pi(s,a) = \frac{e^{f(s,a)}}{\sum_k e^{f(s,a_k)}}$



Pros:	**Support stochastic policy.**
Cons:	Cannot deal with continuous action space. Require predefined features. Linear model has low presentation capacity.

- Neural network policy function (Su *et al.* 2016; Wen *et al.* 2016b)
  - Policy function is approximated by Neural Network.
  - $a = \pi(s) = NN(s, W)$
  - $\phi_l = \sigma({\phi_{l-1}}^T W_l)$
  - ϕ_l is the neural network output of the l-th layer.



Pros:	**Better representation ability. Can work with continuous action space.**
Cons:	Require a lot of data to train, overfitting easily on small dataset.

3.3 Policy Learning: Actor-critic RL

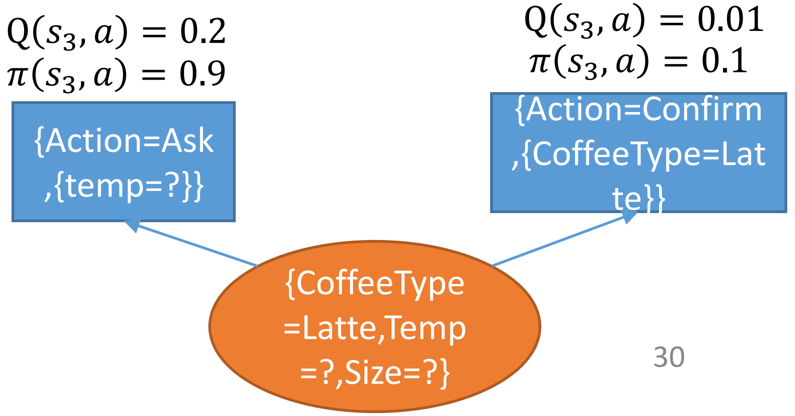
- A Q-function is used as critic and a policy function is used as actor.
  - $Q(s_n, a_n) = E(\sum_{k=0}^{\infty} \gamma^k r_{n+k} \,|s_n, a_n)$
  - $p(a_n|s_n) = \pi(s_n, a_n)$
- Model can be trained with actor-critic (Sutton *et al.* 1999; Konda and Tsitsiklis 1999) algorithm, critic is used as a direction of policy improvement.
  - $\theta = \theta + \alpha \Delta_\theta \log \pi_\theta(s_n, a_n)\, Q(s_n, a_n)$

Pros:	**Has low variance policy gradient estimates, can deal with continuous action space.**
Cons:	Requires compatible function approximators.

	Coffee Shopping Dialogue (X=customer, Y=system)
X_1	I would like a cup of coffee.
Y_1	What coffee would you like?
X_2	What coffee do you serve?
Y_2	We serve Espresso, Americano, Latte, Mocha, etc.
X_3	I would like a cup of Latte.
Y_3	Hot Latte or Iced Latte?
X_4	Hot Latte.
Y_4	What cup size do you want?
X_5	Tall.
R	**<Task-Completion-Feedback>**

Now

$v_3 = Q_\theta(s_3, a_3)$



3.4 Evaluation for Dialogue Policy Learning

- Policy is tested with user simulator (Schatztnann *et al.* 2005)
- [Dialogue Policy Learning method comparison summary.]

Policy Evaluation	
Metrics:	Reward, Success Rate, #dialogue turns
Dataset:	Cambridge Restaurant Domain (TopTable) (Gaši´c and Young 2014); Town Information (Thomson and Young 2010)

Method	Dataset	Reward	Success Rate	# dialogue turns
Gaussian Process	TopTable	11.6 ± 0.4	0.912 ± 0.014	6.6 ± 0.2
Gaussian Process online	TopTable	13.4 ± 0.3	0.968 ± 0.009	6.0 ± 0.1
NAC Jurčíček *et al.* [2011]	TownInfo	3 ± 0.3		
NN based NAC Su *et al.* [2016]	TopTable		0.91	

3.5 Transfer Learning for Dialogue Policy Learning

- When target domain do not have enough data, 3 kinds of transfer learning technique can be used:
  - Linear Model transfer for Q-learning (Genevay and Laroche 2016)
    - Cross domain **data similarity function** is defined.
    - Transfer only diverse data points to target domain.
  - Gaussian Process transfer for Q-learning
    - Transferring mean function and covariance function depends on a **cross domain kernel function**
      - Common slots only (Gasic *et al.* 2014)
      - Similar slot pairs assigned by human (Gašic *et al.* 2013b)
      - Cardinality based slot matching (Gaši´c *et al.* 2015a)
  - Bayesian Committee Machine transfer for Q-learning
    - A BCM is s an ensemble of Gaussian Process policies trained on different datasets in different domains.
    - Gaši´c *et al.* (2015b) proposed an entropy based **cross domain kernel function**.

Pros:	**Simple and efficient.**
Cons:	Requires the source and target domain use the same feature space.

Pros:	**Can deal with different feature space.**
Cons:	Requires common slots and is not scalable.

Pros:	**No common slot is required.**
Cons:	Computationally expensive.

4:Natural Language Generation

- NLG takes system action a_n as input and outputs system response $\boldsymbol{Y}_n$.
- Dialogue action $a_n = \{d_n, \{A_i, V_i\}\}$, d_n is the dialogue act type and A_i, V_i are the name and value of the i-th attribute.
- $\boldsymbol{Y}_n = \{y_1, y_2, \ldots\}$ is the list of words in the n-th response.

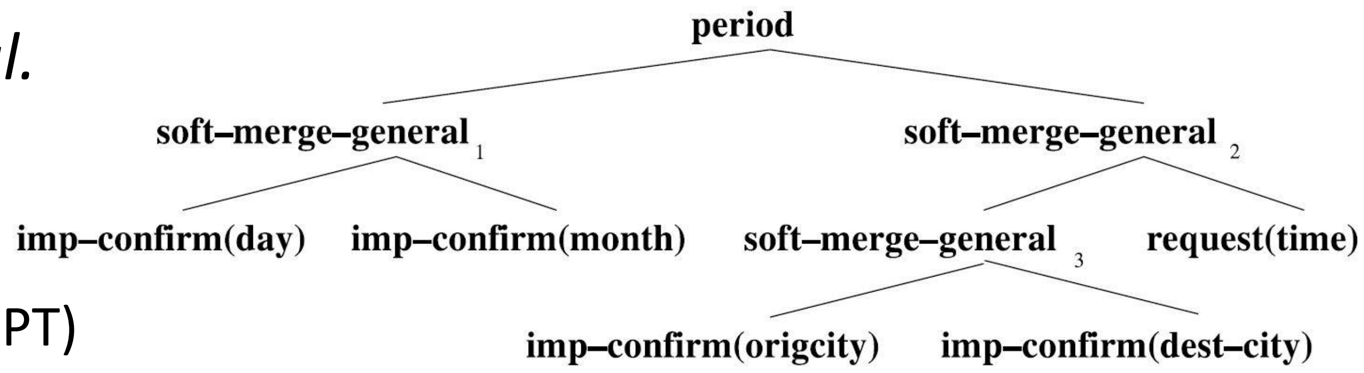
	Coffee Shopping Dialogue (X=customer, Y=system)
$\boldsymbol{X}_1$	I would like a cup of coffee.
$\boldsymbol{Y}_1$	What coffee would you like?
$\boldsymbol{X}_2$	What coffee do you serve?
$\boldsymbol{Y}_2$	We serve Espresso, Americano, Latte, Mocha, etc.
$\boldsymbol{X}_3$	I would like a cup of Latte.
$\boldsymbol{Y}_3$	Hot Latte or Iced Latte?

Now

NLG	
Input:	$\boldsymbol{a}_3$={Action=Ask,{CoffeeType=Latte,Temp=?}},
Output:	$\boldsymbol{Y}_3$ = "Hot Latte or Iced Latte?"

4.1 NLG: Traditional sentence generation

- Language generation (Walker *et al.* 2002; Stent *et al.* 2004) has 2 processes
- Sentence Planning
  - Generate Sentence Planning Tree (SPT)
  - Node is a elementary dialogue act
- Surface Realization
  - Turn SPT to final sentence
  - All children node are merged to become the parent node, with Deep-Syntactic Structure



Sentence Planning Tree for sentence "Leaving on September the 1st. What time would you like to travel from Newark to Dallas?"

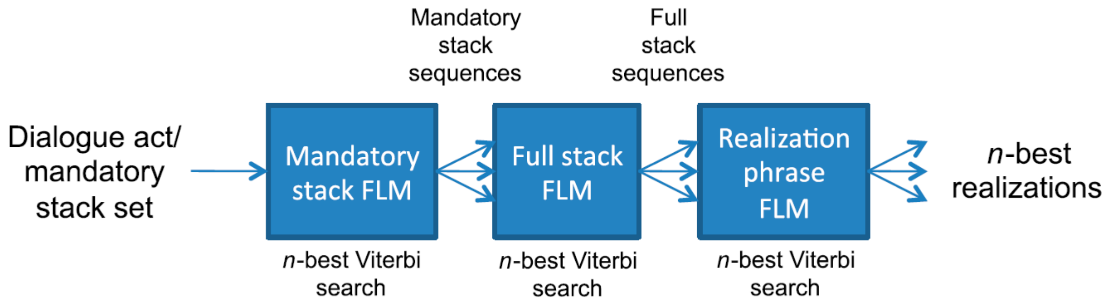
Rule	Sample first argument	Sample second argument	Result
MERGE	You are leaving from Newark.	You are leaving at 5	You are leaving at 5 from Newark
MERGE-GENERAL	What time would you like to leave?	You are leaving from Newark.	What time would you like to leave from Newark?
SOFT-MERGE	You are leaving from Newark	You are going to Dallas	You are traveling from Newark to Dallas
SOFT-MERGE-GENERAL	What time would you like to leave?	You are going to Dallas.	What time would you like to fly to Dallas?

Deep-Syntactic Structure

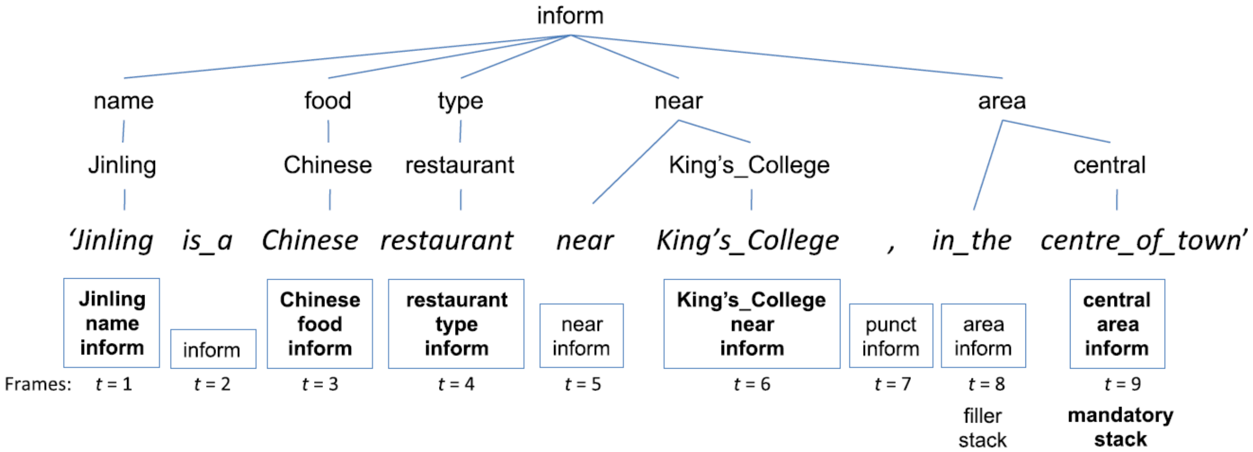
Pros:	**The intermediate representation has clear meaning.**
Cons:	Requires a lot of hand-crafting.

4.2 NLG: Corpus based sentence generation

- Learn generation decisions from data
  - N-gram language model (Oh and Rudnicky 2000)
  - Phase-based NLG based on Factored Language Model (Mairesse and Young 2014)
    - unordered mandatory Stack set-> ordered mandatory Stack sets
    - ordered mandatory Stack set -> full stack sequence
    - full stack sequence -> sentences
  - Template Ranking
    - Log-linear (Angeli *et al.* 2010)
    - SVM (Cortes and Vapnik 1995) is used in Kondadadi *et al.* (2013)



unordered mandatory Stack set-> ordered mandatory Stack set -> full stack sequence

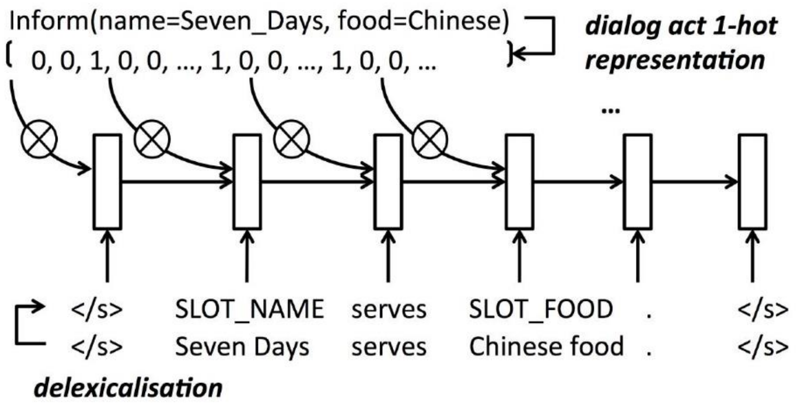


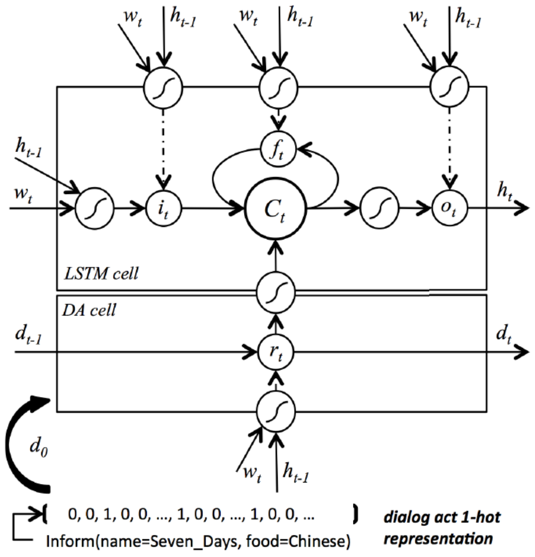
full stack sequence -> sentences

Pros:	**Reduce the human effort and is prone to error.**
Cons:	It requires a predefined feature set.

4.3 NLG: Neural network based sentence generation

- Use neural network to directly generate sentence based system action.
- RNN language models (Mikolov *et al.* 2010)
- RNN+CNN ranking (Wen *et al.* 2015a)
  - $h_t = \sigma(W_t h_{t-1} + U y_t + W_d d_t)$
  - Context vector d_t is controlled with heuristic.
  - CNN sentence ranker is applied at last.
- SC-LSTM (Wen *et al.* 2015b)
  - Context vector is controlled by another gate.
  - $r_t = \sigma(W_t h_{t-1} + U y_t); d_t = r_t \odot d_{t-1}$





Pros:	**Can model context of arbitrary length, and requires minimal human effort.**
Cons:	Requires large amount of data and the parameter can hardly be understood.

4.4 Evaluation of Natural Language Generation

- The generated sentence is compared with ground truth sentence
- NLG method comparison summary.

NLG Evaluation	
Metrics:	BLEU, Slot Error Rate
Dataset:	Tourist Information Dataset (Oh and Rudnicky 2000); Restaurant in San Francisco Dataset (Wen *et al.* 2015a)

Method	Dataset	BLEU	Slot Error Rate
Corpus Based ClassLM Oh and Rudnicky [2000]	Tourist Info	0.06	
Corpus Based Bagel Mairesse and Young [2014]	Tourist Info	0.37	
Corpus Based ClassLM Oh and Rudnicky [2000]	Restaurant in SF	0.627	0.087
Neural Network Based RNNLM+CNN Wen *et al.* [2015a]	Restaurant in SF	0.710	0.015
Neural Network Based sc-LSTM Wen *et al.* [2015b]	Restaurant in SF	0.731	0.0046

4.5 Transfer Learning for Natural Language Generation

- When target domain do not have enough data, 2 kinds of transfer learning technique can be used:
  - Model fine-tuning Transfer for NLG
    - Wen *et al.*, (2013) propose to fine-tune an out of domain model with domain data to achieve transfer learning.
  - Instance Synthesis Transfer for NLG
    - Wen *et al.* (2016c) propose to transfer with synthetic data generation process
    - A sc-lstm model is trained on source domain and fine-tune in target domain.
    - Synthetic data is build by adapting **source** domain instances with new slot-values that appeared only in **target** domain

Pros:	**Easy to implement.**
Cons:	It cannot transfer to new slot-values.

An example realisation in laptop (source) domain:

Zeus 19 is a heavy laptop

delexicalisation

<R-NAME-value> is a <I-WEIGHT-value> <R-TYPE-value>

counterfeiting

<R-NAME-value> is a *<I-FAMILY-value> <R-TYPE-value>

A possible realisation in TV (target) domain:

Apollo 73 is a U76 television

Pros:	**Can benefit slot values not in the source domain.**
Cons:	Might have negative transfer when domain difference is large.

End-to-end dialogue system

- Given user utterance and the dialogue history, the system is to output a response sentence.
- Input:
  - Current user utterance $\boldsymbol{X}_n$
  - Dialogue history $\mathrm{H_x} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_{n-1}\}, H_y = \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \dots, \boldsymbol{Y}_{n-1}\}$
- Output:
  - System response sentence $\boldsymbol{Y}_n$

	Coffee Shopping Dialogue (X=customer, Y=system)
$\boldsymbol{X}_1$	I would like a cup of coffee.
$\boldsymbol{Y}_1$	What coffee would you like?
$\boldsymbol{X}_2$	What coffee do you serve?
$\boldsymbol{Y}_2$	We serve Espresso, Americano, Latte and Mocha.
Now $\boldsymbol{X}_3$	I would like a cup of Latte.
$\boldsymbol{Y}_3$	Hot Latte or Iced Latte?

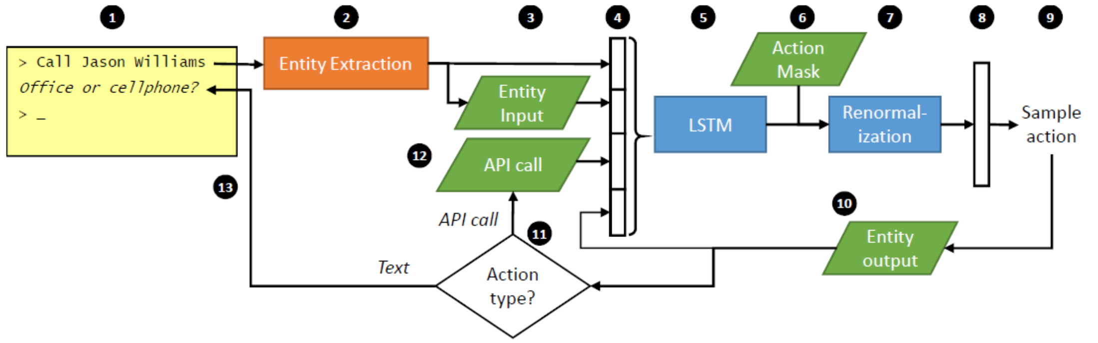
Dialogue System	
Input:	$\boldsymbol{X}_1, \boldsymbol{Y}_1, \boldsymbol{X}_2, \boldsymbol{Y}_2, \boldsymbol{X}_3$
Output:	$\boldsymbol{Y}_3$

End-to-end LSTM Policy network with Answer selection

- Williams and Zweig (2016) propose to jointly model policy network and answer selection process with a LSTM.
- In each round:
  - Hand-crafted SLU+DST+Database map X_n to vector s_n
  - Dialogue Control $a_n = LSTM(s_n)$
  - Answer template selection based on a_n and rules.
  - Template filling.
- Trained with Supervised Learning and RL.

Pros:	**Automatic learn dialogue states.**
Cons:	Many components are still hand-crafted such as SLU, DST, Database.

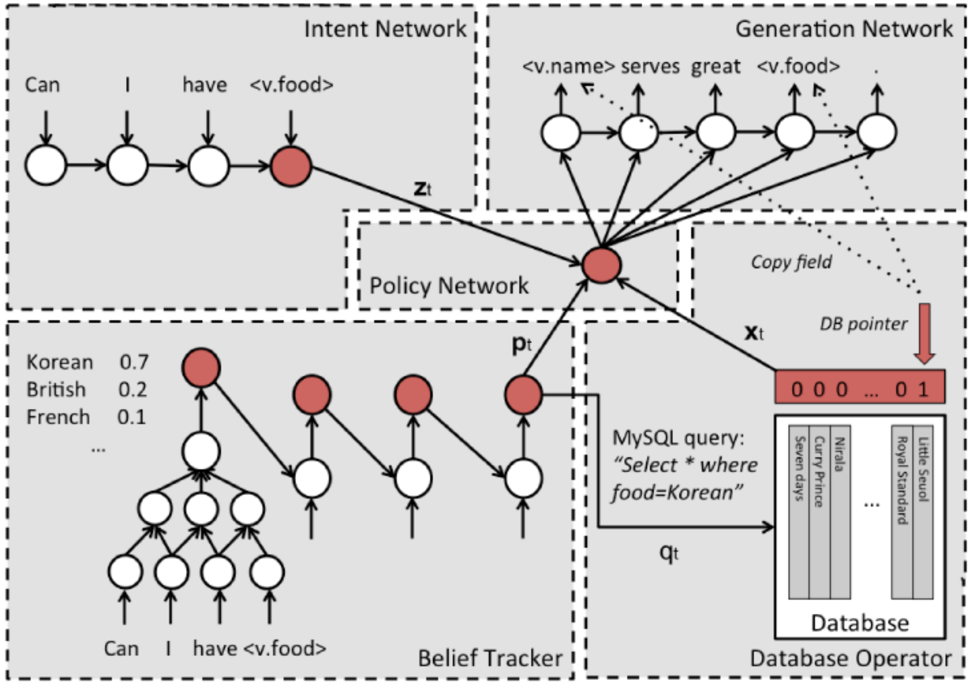
Dialogue System	
Input: $\boldsymbol{X}_1, \boldsymbol{Y}_1, \boldsymbol{X}_2, \boldsymbol{Y}_2, \boldsymbol{X}_3$	I would like a cup of Latte.
Output: $\boldsymbol{Y}_3$	Hot Latte or Iced Latte?



End-to-end training of modular dialogue system

- Wen *et al.* (2016b,a) propose to use trainable modules and train the system end-to-end via gradient descent.
  - SLU: $u_n = LSTM(\boldsymbol{X}_n)$
  - DST: $s_n = RNN(s_{n-1}, CNN(\boldsymbol{X}_n, \boldsymbol{Y}_{n-1}))$
  - Database: $J_n = SQL(s_n)$
  - Dialogue Policy: $a_n = DNN(u_n, s_n, J_n)$
  - NLG: $\boldsymbol{Y}_n = LSTM(a_n)$
- DST module is firstly trained with supervised learning, then all module excluding Database are trained end-to-end.

Dialogue System	
Input: $\boldsymbol{X}_1, \boldsymbol{Y}_1, \boldsymbol{X}_2, \boldsymbol{Y}_2, \boldsymbol{X}_3$	I would like a cup of Latte.
Output: $\boldsymbol{Y}_3$	Hot Latte or Iced Latte?

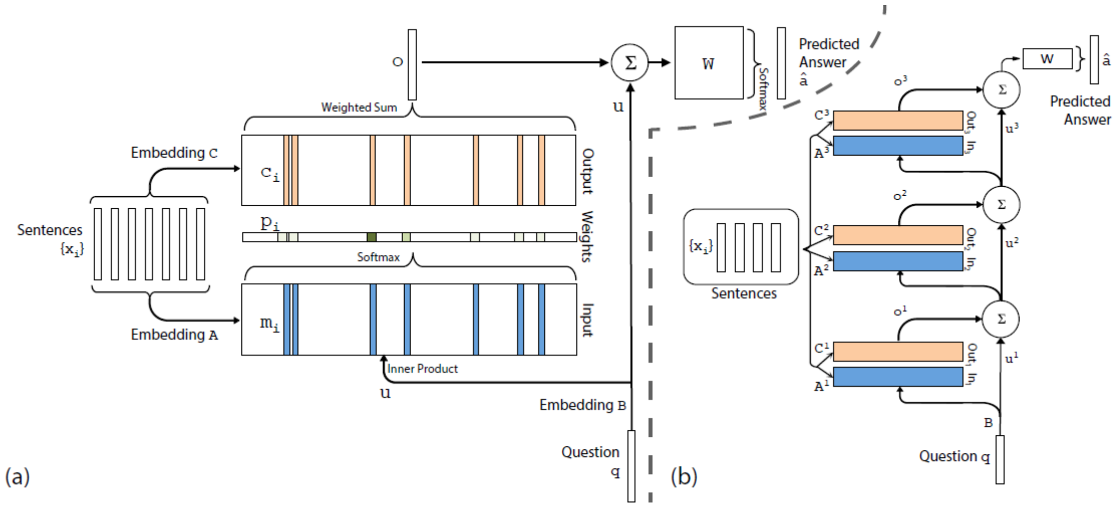


Pros:	**Requires minimal human effort.**
Cons:	The prior knowledge of domain slot value have to be predefine in DST, and DST has to be trained separately.

End-to-end Memory Network based dialogue system

- Bordes and Weston (2016) propose to build goal-oriented dialog system with Memory Networks (Weston et al., 2015a; Sukhbaatar et al., 2015).
- For each round, all **dialogue history** are assumed to be inside the **memory**.
  - $M = \left(\{A\Phi(X_i)\}_i^{n-1}, \{A\Phi(Y_i)\}_i^{n-1}\right)$
  - $q_0 = A\Phi(X_n),$
  - For $h = 0 \rightarrow m$
    - $p_i = \text{softmax}\left(q_h^T m_i\right), o_h = R \sum_i p_i m_i\,,$
    - $q_{h+1} = o_h + q_h$
  - $a = \text{softmax}\left(q_m^T W\Phi(Y_1), \ldots, q_m^T W\Phi(Y_C)\right)$
  - $Y_n = \text{argmax}_Y\, a$

Dialogue System	
Input: $\boldsymbol{X_1, Y_1, X_2, Y_2, X_3}$	I would like a cup of Latte.
Output: $\boldsymbol{Y_3}$	Hot Latte or Iced Latte?



Pros:	**Does not require the modular design of SLU,DST, DPL.**
Cons:	It is difficult to incorporate human knowledge.

Evaluation of End-to-end dialogue system

- Generated sentence will be tested with user simulator or compare with ground truth results.
- Metrics include
  - Slot-Match-Rate for SLU,
  - Task-Success-Rate for dialogue policy,
  - BLEU for language generation,
  - Accuracy for answer selection.
- End-to-end dialogue systems comparison summary.

NLG Evaluation	
Metrics:	Slot-Match-Rate for SLU; Task-Success-Rate (TCR) for Policy; BLEU for NLG (T5 for top 5, and T1 for top 1); Accuracy for answer selection (Acc.T means turn level and Acc.D means dialogue level).
Dataset:	Restaurant domain (Wen *et al.* (2016a) ; Phone call domain (Williams and Zweig 2016).

Method	Dataset	TCR	Slot-Match	BLEU-T5	BLEU-T1	Acc.T	Acc.D
Wen *et al.* [2016b]	Restaurant	0.8382	0.9088	0.2304	0.2369		
Wen *et al.* [2016a]	Restaurant	0.8180	0.6005	0.227	0.2400		
Williams and Zweig [2016]	Phone	0.6900				0.9200	0.4800
Bordes and Weston [2016]	Restaurant					0.9340	0.1970

Tasks VS Techniques in Task-oriented Dialogue System (Detail)

Existing Work ★ Target: Personalization

		Modular Dialogue System					End-to-End Dialogue System	
		1:SLU	2:DST	3: Policy Learning (DPL) General	3: Policy Learning (DPL) Personalized	4:NLG	General	Personalized
None RL	None TL	CRF (Wang and Acero 2006; Raymond and Riccardi 2007) RNN (Yao *et al.* 2013; Mesnil *et al.* 2013, 2015; Liu and Lane 2015) LSTM (Yao *et al.* 2014)	HIS (Young *et al.* 2007) BUDS (Thomson *et al.* 2008) CRF (Lee 2013; Kim and Banchs 2014) RNN (Henderson *et al.* 2014c,d) LSTM (Zilka and Jurcicek 2015)			Conventional NLG (Walker *et al.* 2002; Stent *et al.* 2004) Corpus based (Oh and Rudnicky 2000; Mairesse and Young 2014; Angeli *et al.* 2010; Kondadadi *et al.* 2013) Neural network based (Mikolov *et al.* 2010; Wen *et al.* 2015a; Wen *et al.* 2015b)	RNN based (Wen *et al.* 2016b,a) Memory Network based (Bordes and Weston 2016)	
	TL	Instance based (Tur 2006) Model adaptation (Tür 2005) Parameter Transfer (Yazdani and Henderson)	Feature based transfer (Williams 2013; Ren et al. 2014) Model based transfer (Mrkši´c *et al.* 2015)			Model fine-tuning (Wen *et al.*, 2013) Curriculum learning Transfer (Shi *et al.* 2015) Instance Synthesis Transfer (Wen *et al.* 2016c)		
RL	None TL			Value-based RL (Williams 2008a,b; Young *et al.* 2010; Lefèvre *et al.* 2009; Li *et al.* 2009; Gaši´c and Young 2014; Gaši´c *et al.* 2013a; Daubigney *et al.* 2012b) Policy-based RL (Jurcícek *et al.* 2011; Su *et al.* 2016; Wen *et al.* 2016b) Actor-critic RL (Su et al. 2016; Jurcícek et al. 2011; Misu et al. 2012, 2010)			Policy network with Answer selection (Williams and Zweig 2016)	
	TL			Gaussian Process Transfer (Gasic *et al.* 2014; Gašic *et al.* 2013b; Gaši´c *et al.* 2015a) Bayesian Committee Machine Transfer (Gaši´c *et al.* 2015b)	Linear Model Transfer (Genevay and Laroche 2016). Gaussian Process Transfer (Casanueva et al. 2015) ★			★

Conclusion: Comparison of Techniques in each task

Task	Method	Advantages	Disadvantages
1:SLU	Conditional Random Field	Consider label transition probability.	Use only fixed window size, which is not flexible.
	Recurrent Neural Network	Can model arbitrary long dependency.	Hard to train due to exploding or vanishing gradient problem.
	Long Short Term Memory	Can choose to remember or forget the information stored in its memory.	Require more training data.
2:DST State Represen tation	Hidden Information State Model	Can model arbitrary dependency between any slot value.	Can only retain Top N states.
	Bayesian Update of Dialogue States	Can model probability of all states.	Can model only simple dependency.
2:DST State Tracking	Static Classifier	Do not require previous state, works with any classifiers.	Did not consider state transition, uses hand defined feature extractor.
	Sequential Classifier	Consider state transition probability.	Error in last state might affect the estimation of current state.
3:Policy Learning	Q-learning	Easy to use.	Cannot deal with continuous action space.
	Policy Iteration	Can deal with continuous action space.	Unstable optimization.
	Actor critic	Can deal with continuous action space. Stable optimization.	Require 2 carefully chosen function approximators.

Conclusion: Comparison of Techniques in each task

Task	Method	Advantages	Disadvantages
4:NLG	Traditional sentence generation	The intermediate representation has clear meaning; Easy to encode human knowledge.	Require a lot of hand-crafting; Sensitive to noise.
	Corpus based sentence generation	Reduce the human effort; Prone to error.	Only considered fixed size context; Requires a predefined feature set.
	Neural network based generation	Can model context of arbitrary length; Require minimal human effort.	Require large amount of training data; Hard to understand, hard to encode human knowledge.
End-to-end	End-to-end LSTM Policy network with Answer selection	Automatic dialogue state learning.	Many components are still hand crafted and cannot be trained, such as SLU, DST, Database.
	End-to-end training of modular dialogue system	Require minimal human effort.	Domain knowledge such as Number of slots, possible slot value have to be predefine. DST has to be trained separately, requiring more data labelling.
	Memory Network based end-to-end dialogue system	No information about Slot and slot value is required.	It is difficult to incorporate human knowledge.

Conclusion: Model vs Task summary table

- Static classifiers and sequential classifiers can be used for many tasks.
- End-to-end models are developing fast.

Models	Task	Formulation
Grid	SLU,DST,Policy	$Q(s,a) = \text{Table}(\phi(s), \phi(a))$
Linear	SLU,DST,Policy	$y = \mathbf{w}^T \mathbf{x}$
Gaussian Process	Policy	$Q^\pi(s,a) \sim \mathcal{GP}(m(s,a), k((s,a),(s,a)))$
MLP	SLU,DST,Policy	$y_0 = x$ $y_n = \sigma(\mathbf{W} y_{n-1})$
CRF	SLU,DST	$P(y_n \mid x_n, y_{n-1}) = \frac{1}{Z} exp(p(y_n \mid y_{n-1}) + p(y_n \mid x_n))$
RNN	SLU,DST,NLG	$\mathbf{h}_n = \sigma(\mathbf{W}_h \mathbf{h}_{n-1} + \mathbf{U}\mathbf{x}_n + \mathbf{b}_h)$ $p(y_n \mid x_n, x_{<n}) = \frac{1}{Z(\mathbf{h}_n)} \sigma(\mathbf{w}_{y_n}^T \mathbf{h}_n + b_{y_n})$
LSTM	SLU,DST,NLG	$\begin{bmatrix} \mathbf{i}_n \\ \mathbf{o}_n \\ \mathbf{f}_n \\ \hat{\mathbf{c}}_n \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \mathbf{W} \begin{bmatrix} \mathbf{h}_{n-1} \\ \mathbf{x}_n \end{bmatrix}$ $\mathbf{c}_n = \mathbf{f}_n \odot \mathbf{c}_{n-1} + \mathbf{i}_n \odot \hat{\mathbf{c}}_n$ $\mathbf{h}_n = \mathbf{o}_n \odot tanh(\mathbf{c}_n)$ $p(y_n \mid x_n, x_{<n}) = \frac{1}{Z(\mathbf{h}_n)} \sigma(\mathbf{w}_{y_n}^T \mathbf{h}_n + b_{y_n})$
MemoryNN	SLU+DST+Policy	$\mathbf{M} = (\mathbf{A}\Phi(\mathbf{X}_1), \mathbf{A}\Phi(\mathbf{Y}_1), \cdots, \mathbf{A}\Phi(\mathbf{X}_{n-1}), \mathbf{A}\Phi(\mathbf{Y}_{n-1}))$ $q_0 = \mathbf{A}\Phi(\mathbf{X}_n)$ $p_i = \text{Softmax}(\mathbf{q}_h^T \mathbf{m}_i)$ $\mathbf{o}_h = \mathbf{R}\Sigma_i p_i \mathbf{m}_i$ $\mathbf{q}_{h+1} = \mathbf{o}_h + \mathbf{q}_h$ $\mathbf{a} = \text{Softmax}(\mathbf{q}_h^T \mathbf{W}\Phi(\mathbf{Y}_1), \cdots, \mathbf{q}_h^T \mathbf{W}\Phi(\mathbf{Y}_C))$

Conclusion: Dataset vs Evaluation Summary Table

- Most evaluation are for separate modules.
- Online evaluation are done with user simulators, which is not diverse and could not reflect real life complexity.
- There are no automatic online end-to-end evaluation metrics, every metrics require human labelling.

Dataset	Task	Evaluation Metrics
ATIS Dahl *et al.* [1994]	SLU	Accuracy, F1
TouristInfo Bonneau-Maynard *et al.* [2005]	SLU, NLG	Accuracy, F1, BLEU
DARPA Communicator Walker *et al.* [2001]	SLU	Accuracy, F1
DSTC 1 Williams *et al.* [2013]	DST	Accuracy, L2
DSTC 2 Henderson *et al.* [2014a]	DST	Accuracy, L2
DSTC 3 Henderson *et al.* [2014b]	DST	Accuracy, L2
TownInfo Thomson and Young [2010]	DST, Policy	Accuracy, L2, Reward, Success Rate, #Dialog Turns
Cambridge Restaurant Gašić and Young [2014]	Policy, NLG	Reward, Success Rate, #Dialog Turns, BLEU
SF Restaurant Gašić *et al.* [2015b]	Policy, NLG	Reward, Success Rate, #Dialog Turns, BLEU
SF Hotel Gašić *et al.* [2015b]	Policy, NLG	Reward, Success Rate, #Dialog Turns, BLEU
L11 Gašić *et al.* [2015b]	Policy, NLG	Reward, Success Rate, #Dialog Turns, BLEU

Future directions

- End-to-end dialogue systems which requires less human knowledge.
- Dialogue personalization.
- Knowledge transfer between dialogue domains.
- Learning to expend dialogue domains in online setting with Reinforcement Learning.

- Research conducted:
  - Mo, K., Li, S., Zhang, Y., Li, J., & Yang, Q. (2016). Personalizing a Dialogue System with Transfer Learning. *arXiv preprint arXiv:1610.02891*.

Reference

Survey of Task-oriented Dialogue System

SLU-CRF

- John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001.
- Ye-Yi Wang and Alex Acero. Discriminative models for spoken language understanding. In *INTER-SPEECH*, 2006.
- Christian Raymond and Giuseppe Riccardi. Generative and discriminative algorithms for spoken language understanding. In *INTERSPEECH*, pages 1605–1608, 2007.
- Puyang Xu and Ruhi Sarikaya. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 78–83. IEEE, 2013.

SLU-RNN

- Christoph Goller and AndreasKuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE, 1996.
- Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. Recurrent neural networks for language understanding. In *INTERSPEECH*, pages 2524–2528, 2013.
- Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. Investigation of recurrent-neuralnetwork architectures and learning methods for spoken language understanding. In *INTER-SPEECH*, pages 3771–3775, 2013.
- Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539, 2015.

SLU-LSTM

- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. Spoken language understanding using long short-term memory neural networks. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 189–194. IEEE, 2014.

SLU-TL

- Gokhan Tur. Multitask learning for spoken language understanding. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.
- Gökhan Tür. Model adaptation for spoken language understanding. In *ICASSP (1)*, pages 41–44. Citeseer, 2005.

DST-State Representation

- Steve Young, Jost Schatzmann, Karl Weilhammer, and Hui Ye. The hidden information state approach to dialog management. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–149. IEEE, 2007.
- *Murphy, Kevin (2002). Dynamic Bayesian Networks: Representation, Inference and Learning. UC Berkeley, Computer Science Division.*
- Blaise Thomson, Jost Schatzmann, and Steve Young. Bayesian update of dialogue state for robust dialogue systems. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4937–4940. IEEE, 2008.

DST-Sequential tracking-CRF

- Sungjin Lee. Structured discriminative model for dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 442–451, 2013.
- Seokhwan Kim and Rafael E Banchs. Sequential labeling for tracking dynamic dialog states. In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 332, 2014.

- Seokhwan Kim and Rafael E Banchs. Sequential labeling for tracking dynamic dialog states. In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 332, 2014.

DST-Sequential tracking-RNN+LSTM

- Matthew Henderson, Blaise Thomson, and Steve Young. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 360–365. IEEE, 2014.
- Lukas Zilka and Filip Jurcicek. Incremental lstm-based dialog state tracker. In *2015 IEEEWorkshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 757–762. IEEE, 2015.

DST-TL

- Jason Williams. Multi-domain learning and generalization in dialog state tracking. In *Proceedings of SIGDIAL*, volume 62, 2013.
- Hang Ren, Weiqun Xu, and Yonghong Yan. Markovian discriminative modeling for cross-domain dialog state tracking. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 342–347. IEEE, 2014.
- Nikola Mrkši´c, Diarmuid O Séaghdha, Blaise Thomson,Milica Gaši´c, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Multi-domain dialog state tracking using recurrent neural networks. *arXiv preprint arXiv:1506.07190*, 2015.

Policy-Value Based

- Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. Vol. 1. No. 1. Cambridge: MIT press, 1998.
- Christopher John Cornish HellabyWatkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge England, 1989.
- Jason D Williams. The best of both worlds: unifying conventional dialog systems and pomdps. In *INTERSPEECH*, pages 1173–1176, 2008.
- Steve Young, Jost Schatzmann, Karl Weilhammer, and Hui Ye. The hidden information state approach to dialog management. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–149. IEEE, 2007.
- Fabrice Lefèvre, Milica Gaši´c, F Jurˇcíˇcek, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and S Young. k-nearest neighbor monte-carlo control algorithm for pomdp-based dialogue systems. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 272–275. Association for Computational Linguistics, 2009.
- Lihong Li, Jason D Williams, and Suhrid Balakrishnan. Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *INTERSPEECH*, pages 2475–2478, 2009.
- Milica Gaši´c and Steve Young. Gaussian processes for pomdp-based dialoguemanager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40, 2014.
- M Gaši´c, Catherine Breslin, Matthew Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, and Steve Young. On-line policy optimisation of bayesian spoken dialogue systems via human interaction. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8367–8371. IEEE, 2013.
- Lucie Daubigney, Matthieu Geist, and Olivier Pietquin. Off-policy learning in large-scale pomdpbased dialogue systems. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4989–4992. IEEE, 2012.

Policy-Policy Based

- Ronald JWilliams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256, 1992.
- Filip Jurcícek, Blaise Thomson, and Steve Young. Natural actor and belief critic: Reinforcement algorithm for learning parameters of dialogue systems modelled as pomdps. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):6, 2011.
- Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689*, 2016.
- Tsung-HsienWen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

Policy-Actor Critic

- Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In NIPS, volume 99, pages 1057–1063, 1999.
- Vijay R Konda and John N Tsitsiklis. On actor-critic algorithms. SIAM journal on Control and Optimization, 42(4):1143–1166, 2003.

Policy-TL

- Aude Genevay and Romain Laroche. Transfer learning for user adaptation in spoken dialogue systems. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 975–983, 2016.
- Milica Gasic, DonghoKim, Pirros Tsiakoulis, Catherine Breslin,Matthew Henderson,Martin Szummer, Blaise Thomson, and Steve J. Young. Incremental on-line adaptation of POMDP-based dialogue managers to extended domains. In *Proceedings of the 15th Annual Conference of the International Speech Communication Association*, pages 140–144, 2014.
- Milica Gašic, Catherine Breslin, Matthew Henderson, Dongho Kim,Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, and Steve Young. POMDP-based dialoguemanager adaptation to extended domains. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2013.
- M Gaši´c, Dongho Kim, Pirros Tsiakoulis, and Steve Young. Distributed dialogue policies for multidomain statistical dialogue management. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5371–5375. IEEE, 2015.
- M Gaši´c, N Mrkši´c, PH Su, David Vandyke, Tsung-HsienWen, and S Young. Policy committee for adaptation in multi-domain spoken dialogue systems. 2015.

NLG-Traditional

- Marilyn A Walker, Owen C Rambow, and Monica Rogati. Training a sentence planner for spoken dialogue using boosting. *Computer Speech & Language*, 16(3):409–433, 2002.
- Amanda Stent, Rashmi Prasad, and Marilyn Walker. Trainable sentence planning for complex information presentation in spoken dialog systems. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 79. Association for Computational Linguistics, 2004.

NLG-Corpus based

- Alice H Oh and Alexander I Rudnicky. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*, pages 27–32. Association for Computational Linguistics, 2000.
- François Mairesse and Steve Young. Stochastic language generation in dialogue using factored language models. *Computational Linguistics*, 2014.
- Gabor Angeli, Percy Liang, and Dan Klein. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512. Association for Computational Linguistics, 2010.
- Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- Ravi Kondadadi, Blake Howald, and Frank Schilder. A statistical nlg framework for aggregated planning and realization. In *ACL (1)*, pages 1406–1415, 2013.

NLG-Neural Network

- Christoph Goller and AndreasKuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE, 1996.
- Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernock`y, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.
- Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan ˇCernock`y, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531. IEEE, 2011.
- Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *arXiv preprint arXiv:1508.01755*, 2015.
- Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

NLG-TL

- Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. Cognition, 48, 781–799
- Tsung-Hsien Wen, Aaron Heidel, Hung-yi Lee, Yu Tsao, and Lin-Shan Lee. Recurrent neural network based languagemodel personalization by social network crowdsourcing. In *INTERSPEECH*, pages 2703–2707, 2013.
- Yangyang Shi, Martha Larson, and Catholijn M Jonker. Recurrent neural network language model adaptation with curriculum learning. *Computer Speech & Language*, 33(1):136–154, 2015.
- Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. Multi-domain neural network language generation for spoken dialogue systems. *arXiv preprint arXiv:1603.01232*, 2016.

End-to-end

- Jason D Williams and Geoffrey Zweig. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*, 2016.
- Tsung-HsienWen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.
- Tsung-HsienWen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. Conditional generation and snapshot learning in neural dialogue systems. *arXiv preprint arXiv:1606.03352*, 2016.

End-to-end

- Weston, J., Chopra, S., and Bordes, A. (2015a). Memory networks. Proceedings of ICLR.
- Sukhbaatar, S., Szlam, A., Weston, J., and Fergus, R. (2015). End-to-end memory networks. Proceedings of NIPS.
- Antoine Bordes and Jason Weston. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*, 2016.

Tasks VS Techniques in Task-oriented Dialogue System (Overview)

Existing Work ★ Target: Personalization

Task	Technique	Papers
1:SLU	SL	CRF (Wang and Acero 2006; Raymond and Riccardi 2007); RNN (Yao *et al.* 2013; Mesnil *et al.* 2013, 2015; Liu and Lane 2015); LSTM (Yao *et al.* 2014)
	SL+TL	Instance based (Tur 2006); Model adaptation (Tür 2005); Parameter Transfer (Yazdani and Henderson)
2:DST	SL	HIS (Young *et al.* 2007); BUDS (Thomson *et al.* 2008); CRF (Lee 2013; Kim and Banchs 2014); RNN (Henderson *et al.* 2014c,d); LSTM (Zilka and Jurcicek 2015)
	SL+TL	Feature based transfer (Williams 2013; Ren et al. 2014); Model based transfer (Mrkši´c *et al.* 2015)
3:Policy	RL	Value-based RL (Williams 2008a,b; Young *et al.* 2010; Lefèvre *et al.* 2009; Li *et al.* 2009; Gaši´c and Young 2014; Gaši´c *et al.* 2013a; Daubigney *et al.* 2012b); Policy-based RL (Jurcícek *et al.* 2011; Su *et al.* 2016; Wen *et al.* 2016b); Actor-critic RL (Su et al. 2016; Jurcícek et al. 2011; Misu et al. 2012, 2010)
	RL+TL	Gaussian Process Transfer (Gasic *et al.* 2014; Gašic *et al.* 2013b; Gaši´c *et al.* 2015a); Bayesian Committee Machine Transfer (Gaši´c *et al.* 2015b)
3:Personalized Policy	RL+TL ★	Linear Model Transfer (Genevay and Laroche 2016); Gaussian Process Transfer (Casanueva et al. 2015)
4:NLG	SL	Conventional NLG (Walker *et al.* 2002; Stent *et al.* 2004); Corpus based (Oh and Rudnicky 2000; Mairesse and Young 2014; Angeli *et al.* 2010; Kondadadi *et al.* 2013); Neural network based (Mikolov *et al.* 2010; Wen *et al.* 2015a; Wen *et al.* 2015b)
	SL+TL	Model fine-tuning (Wen *et al.*, 2013); Curriculum learning Transfer (Shi *et al.* 2015); Instance Synthesis Transfer (Wen *et al.* 2016c)
End-to-end	SL	RNN based (Wen *et al.* 2016b,a); Memory Network based (Bordes and Weston 2016)
	RL	Policy network with Answer selection (Williams and Zweig 2016)
Personalized End-to-end	RL+TL ★	

Note: Each task is corresponding to a module in the flowchart.

Spoken Language Understanding Model comparison

Method	Advantages	Disadvantages
Conditional Random Field	Consider label transition probability.	Use only fixed window size, which is not flexible.
Recurrent Neural Network	Can model arbitrary long dependency.	Hard to train due to exploding or vanishing gradient problem.
Long Short Term memory	Can choose to remember or forget the information stored in its memory.	Requires more training data.

Back to Evaluation.

Transfer Model for Spoken Language Understanding: Model comparison

Method	Advantages	Disadvantages
Model adaptation for Spoken Language Understanding	Easy to use.	Source and target domain have to use the same kind of model.
Instance based transfer for Spoken Language Understanding	Works with any classifiers and do not require predefine instance similarity.	The source domain and the target model need to be trained for multiple times before converging, which is time consuming.
Parameter Transfer for Spoken Language Understanding	Works in zeros-shot setting.	Requires extra data for learning word embedding

Dialogue State Tracking Model comparison

	Method	Advantages	Disadvantages
State Representation	Hidden Information State Model	Can model arbitrary dependency between any slot value pair.	Can only track probability of Top N states.
	Bayesian Update of Dialogue States	Can model probability of all states.	Can model only simple dependency.
State Tracking	Static Classifier	Do not require previous state, works with any classifiers.	Do not consider state transition, use hand defined feature extractor.
	Sequential Classifier	Consider state transition probability.	Error in last state might affect the estimation of current state.

Back to Evaluation.

Sequential State Tracker Comparison

Method	Advantages	Disadvantages
Conditional Random Field	Models state transition probability.	Considers only fixed window size, which is not flexible.
Recurrent Neural Network	Can model arbitrary long dependency.	Hard to train due to exploding or vanishing gradient problem.
Long Short Term Memory	Can choose to remember or forget the information stored in its memory.	Requires more training data.

Back to Evaluation.

Transfer Model for Dialogue State Tracking : Model Comparison

Method	Advantages	Disadvantages
Feature based transfer for Dialogue State Tracking	Easy to implement.	Require careful feature design and lots of human effort.
Model based transfer for Dialogue State Tracking	Able to transfer between fine granularity slots.	Can only work on delexicalised data.

Reinforcement Learning Framework for Dialogue Policy Learning

- Value-based RL
  - Grid based Q-function
  - Linear model Q-function
  - Gaussian Process based Q-function
  - Neural Network based Q-function
- Policy-based RL
  - Softmax Policy function
  - Neural Network Policy function
- Actor-critic RL

Dialogue Policy Learning Framework comparison

Method	Advantages	Disadvantages
Q-learning	Easy to use.	Cannot deal with continuous action space.
Policy Iteration	Can deal with continuous action space.	Unstable optimization.
Actor critic	Can deal with continuous action space. Stable optimization.	Require 2 carefully chosen function approximator.

Back to Evaluation.

Value-based model comparison

Method	Advantages	Disadvantages
Grid based Q-function	Easy to implement.	Hard to decide number of clusters. Large quantization error.
Linear model Q-function	Less quantization error, training and testing are fast.	Require predefined features. Linear model has low representation capacity.
Gaussian Process based Q-function	None-parametric model, avoid the limitation of a human selected basis. Can be trained with small sample.	Slow, high computational complexity and can only process small dataset.
Neural Network based Q-function	Better representation ability.	Require a lot of data to train, overfitting easily on small dataset.

Back to Evaluation.

Policy-based model comparison

Method	Advantages	Disadvantages
Softmax policy function	Probabilistic representation, support stochastic policy.	Cannot deal with continuous action space. Require predefined features. Linear model has low representation capacity.
Neural Network based policy function	Better representation ability. Can work with continuous action space.	Require a lot of data to train, overfitting easily on small dataset.

Back to Evaluation.

Transfer Learning for Dialogue Policy: Model comparison

Method	Advantages	Disadvantages
Linear Model transfer for Q-learning	Simple and efficient, and it can support very large training data.	Source domain and target domain must have the same feature space
Gaussian Process transfer for Q-learning	Do not require the source and target to share the same feature space; None-parametric method, which is more flexible and works well on small data.	Assume the existence of common slots in source and target domain; Computationally expensive and could not support very large training set.
Bayesian Committee Machine transfer for Q-learning	Do not assume the existence of common slots in source and target domains.	Computationally expensive and could not support very large training set.

Natural Language Generation Model comparison

Method	Advantages	Disadvantages
Traditional sentence generation	The intermediate representation has clear meaning; Easy to encode human knowledge.	Require a lot of hand-crafting; Sensitive to noise.
Corpus based sentence generation	Reduce the human effort; Prone to error.	Only modelled fixed size context; Require a predefined feature set.
Neural network based generation	Can model context of arbitrary length; Require minimal human effort.	Require large amount of training data; Hard to understand, hard to encode human knowledge.

Back to Evaluation.

Transfer Learning for Natural Language Generation: Model comparison

Method	Advantages	Disadvantages
Model fine-tuning Transfer for Natural Language Generation	Easy to implement. Can be applied on a trained model.	Cannot benefit slot-values that did not appeared in source domain.
Instance Synthesis Transfer for Natural Language Generation	Can benefit slot-values that did not appeared in source domain.	Might have negative transfer when there is large expressions difference between across domain.

End-to-end Dialogue System comparison

Method	Advantages	Disadvantages
End-to-end LSTM Policy network with Answer selection	Automatic dialogue state learning.	Many components are still hand crafted and cannot be trained, such as SLU, DST, Database.
End-to-end training of modular dialogue system	Require minimal human effort.	Domain knowledge such as slot value have to be predefine. DST has to be trained separately, requiring additional data.
Memory Network based end-to-end dialogue system	Does not require the modular design of SLU,DST, DPL.	It is difficult to incorporate human knowledge inside.

Back to Evaluation.