Dialog Systems

Modern Perspective

by

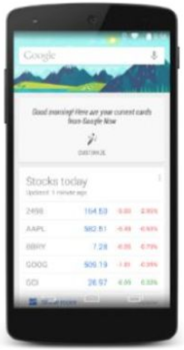
Valentin Malykh
valentin.malykh@phystech.edu

Overview

- Introduction and Background
- Modular Dialogue System
  - Spoken/Natural Language Understanding (SLU/NLU)
  - Dialogue State Tracking (DST)
  - Dialogue Policy
  - Natural Language Generation (NLG)
- End-to-End Learning for Dialogue Systems
- Conclusion

Examples

Apple Siri (2011)



Google Now (2012)

Microsoft Cortana (2014)

Amazon Alexa/Echo (2014)

Facebook M & Bot (2015)

Google Home (2016)

Dialogue System

Task-Oriented

- Personal assistant, achieve a certain task
- Combination of rules and statistical components
  - POMDP for spoken dialog systems (Williams and Young, 2007)
  - Learning End-to-End Goal-oriented Dialog (Antoni and Weston, 2016)
  - An End-to-End Trainable Task-oriented Dialogue System (Wen el al., 2016)

Chit-Chat

- No specific goal, focus on conversation flow
- Work using variants of seq2seq model
  - A Neural Conversation Model (Vinyals and Le, 2015)
  - Deep Reinforcement Learning for Dialogue Generation (Li et al., 2016)
  - Conversational Contextual Cues: The Case of Personalization & History for Response Ranking (AIRfou et al., 2016)

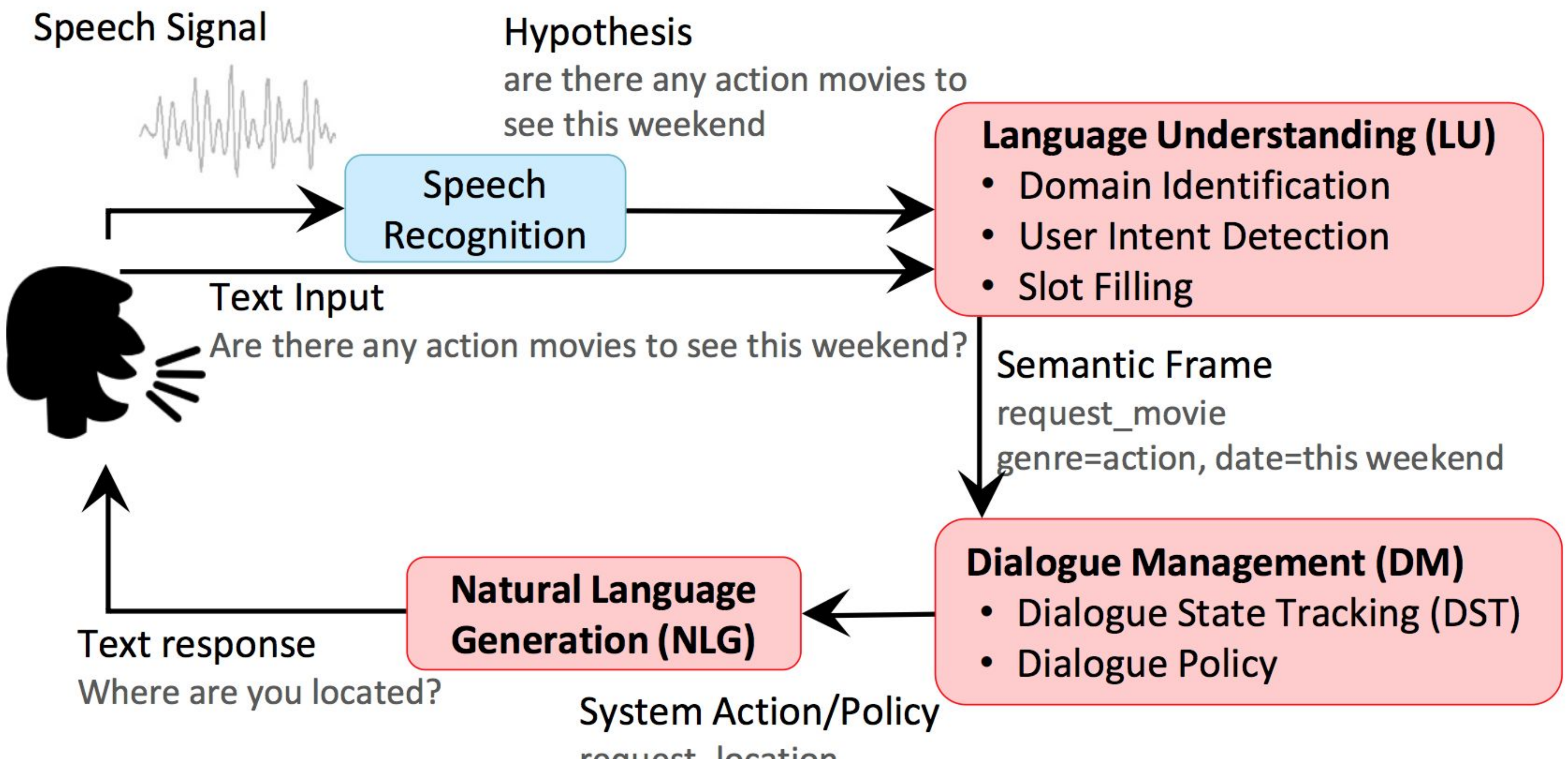
Dialog System Pipeline

Natural Language Understanding

Dialog Management

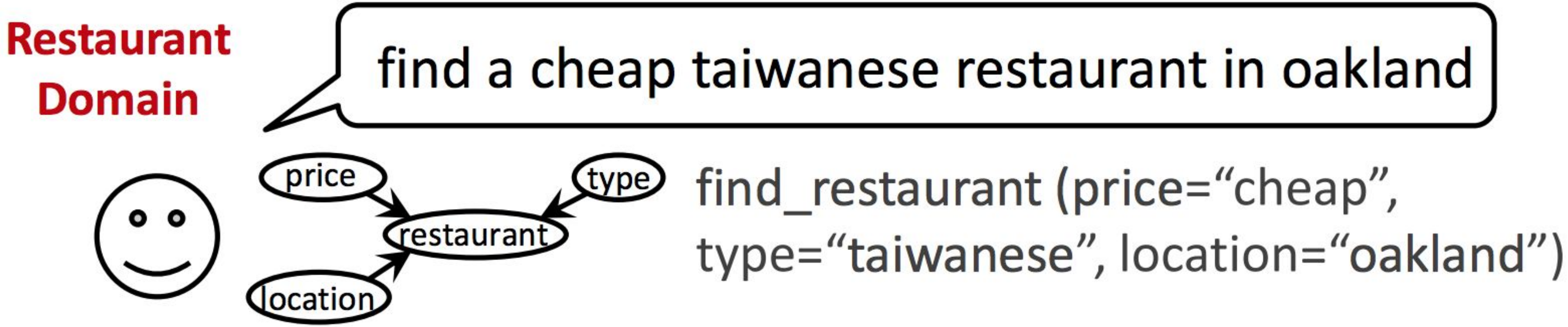
Natural Language Generation

Task-Oriented Dialogue System



Semantic Frame Representation

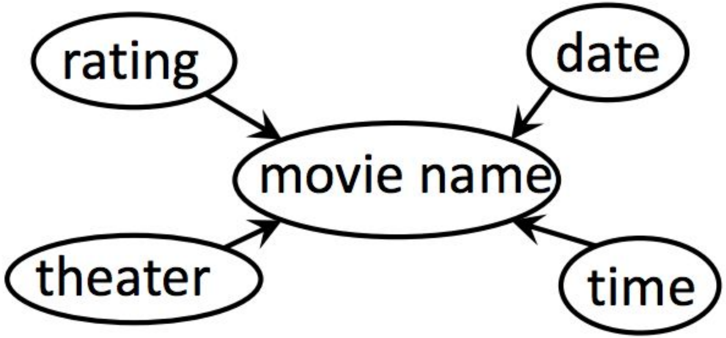
- Requires a domain ontology
- Contains core content (intent, a set of slots with fillers)



Database / Ontology

Domain-specific table

Target and attributes



Functionality

Information access

Finding the specific entries from the table

E.g. available theater, movie rating, etc.

Task completion

Finding the row that satisfies the constraints

Dialogue Schema

Slot: domain-specific attributes

Columns from the table

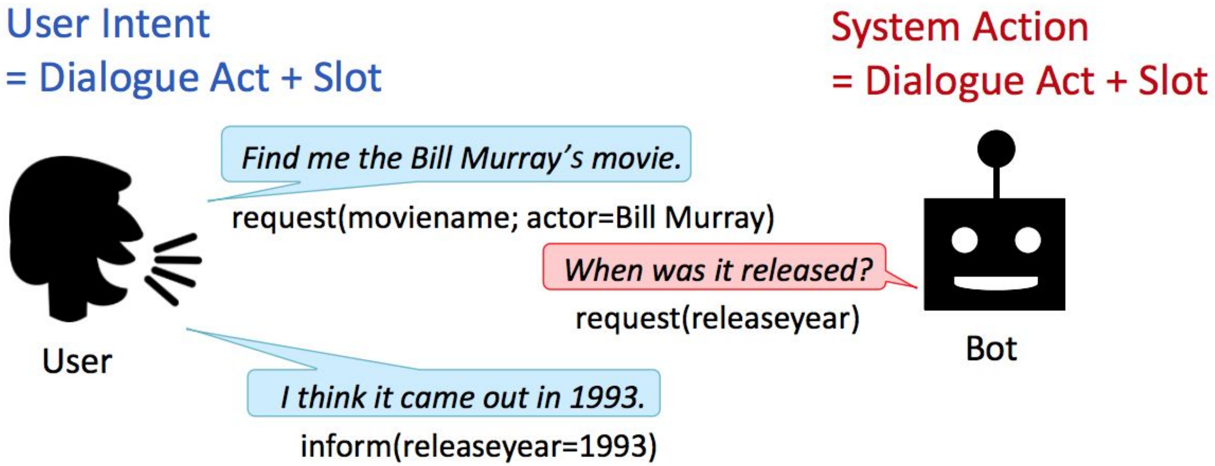
e.g. theater, date

Dialogue Schema

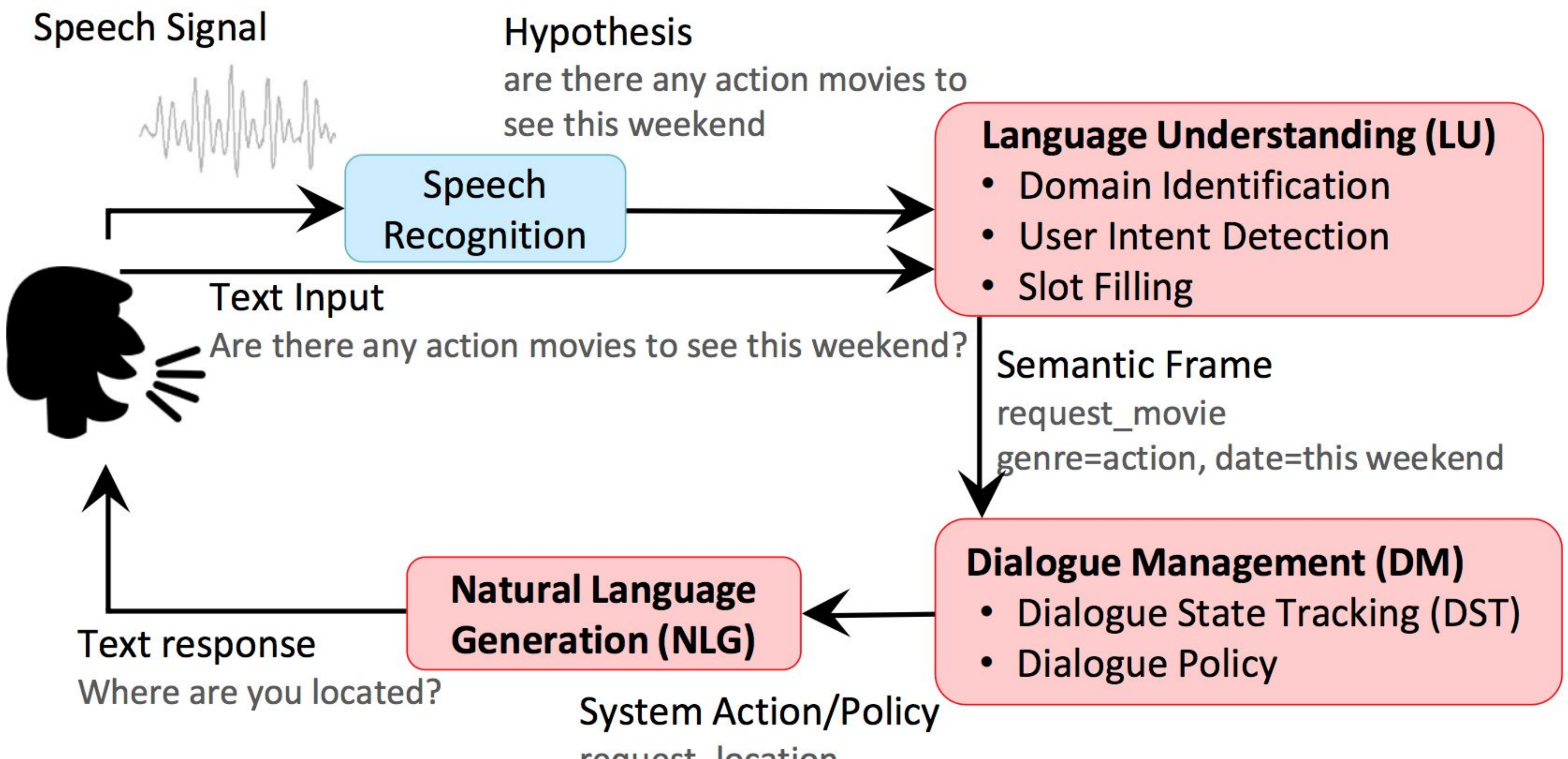
Dialogue Act: inform, request, confirm (system only)

Task-specific action (e.g. book_ticket)

Others (e.g. thanks)



Task-Oriented Dialogue System

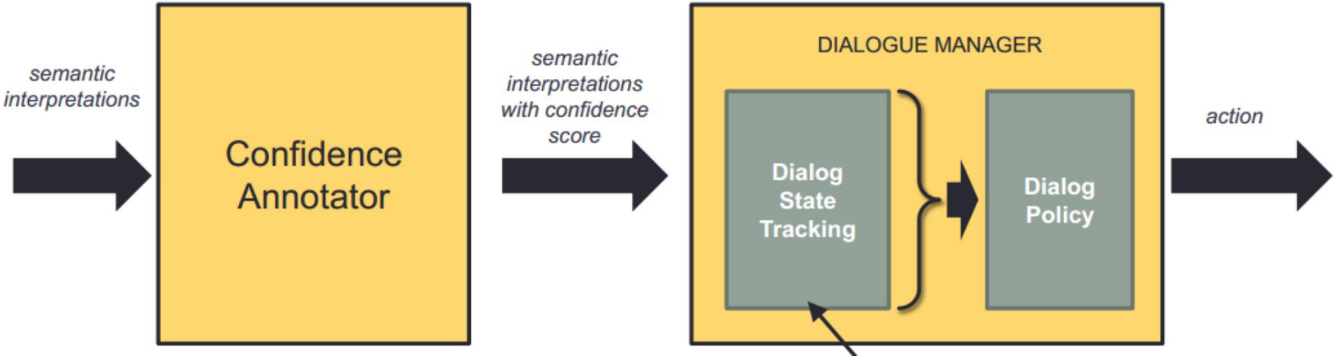


Dialog Management

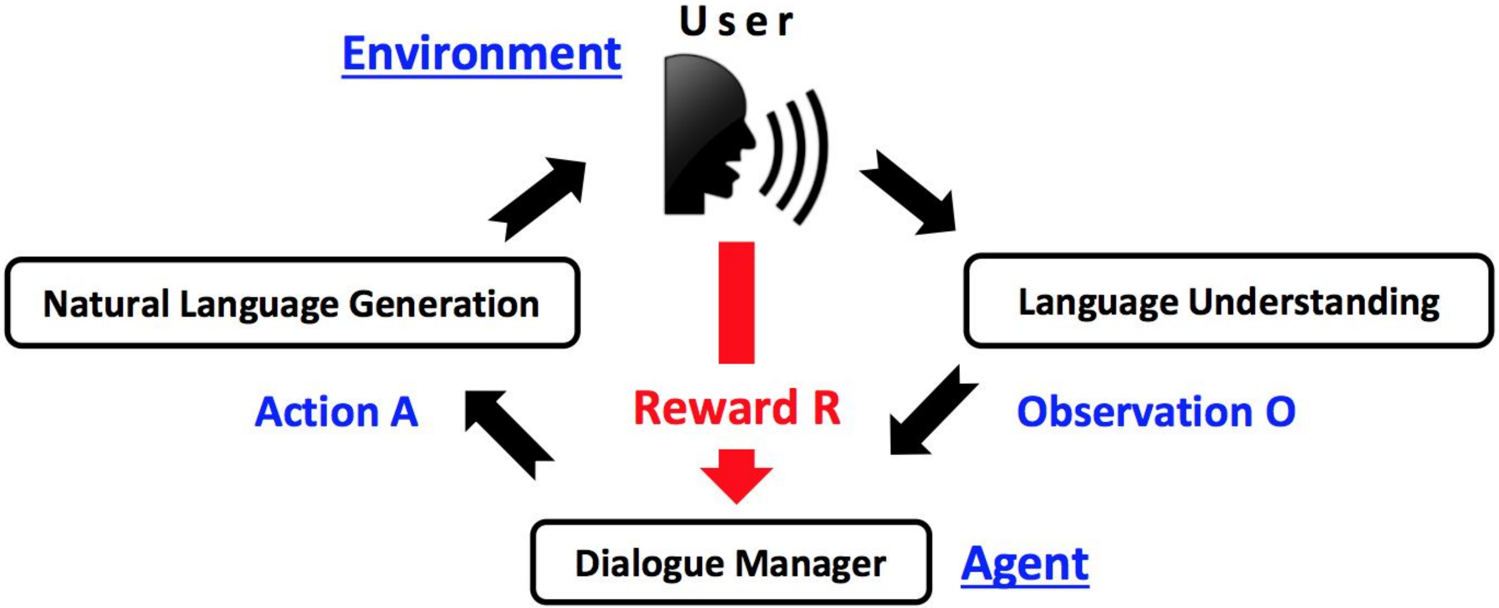
sample problem

S: where would you like to fly from?
U: `[Boston/0.45]; [Austin/0.30]`
S: sorry, did you say you wanted to fly from Boston?
U: `[No/0.37] + [Aspen / 0.7]`

Updated belief = ?
`[Boston/?; Austin/?; Aspen/?]`



Dialog Policy Optimization



Optimized dialogue policy selects the best action that can maximize the future reward.
Correct rewards are a crucial factor in dialogue policy training

Reward for Reinforcement Learning

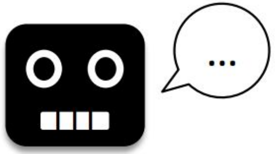
- Dialogue is a special RL task
  - Human involves in interaction and rating (evaluation) of a dialogue
  - Fully human-in-the-loop framework
- Rating: correctness, appropriateness, and adequacy

- Expert rating	high quality, high cost
- User rating	unreliable quality, medium cost
- Objective rating	Check desired aspects, low cost

Reward for Reinforcement Learning

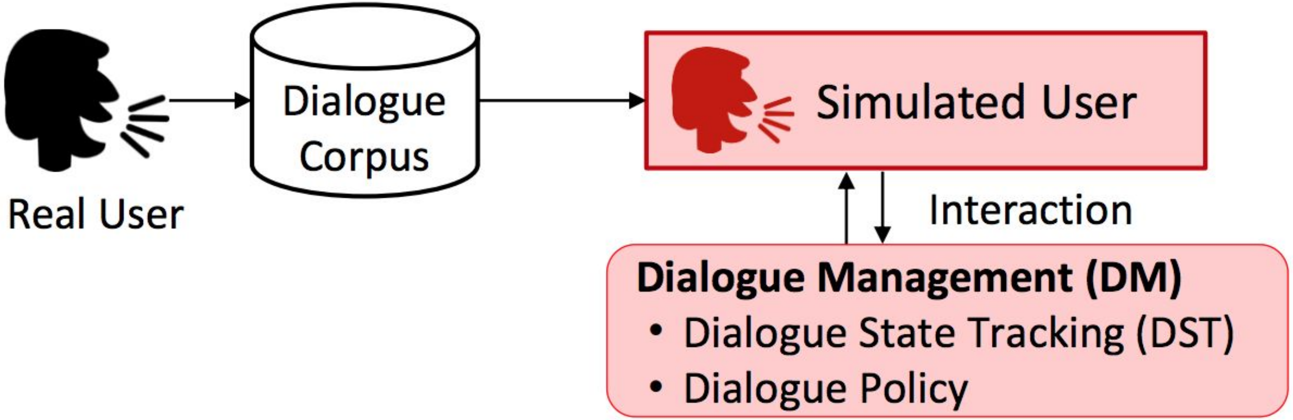
- Typical Reward Function
  - per turn penalty -1
  - Large reward at completion if successful

- Typically requires domain knowledge
  - Simulated user
  - Paid users (Amazon Mechanical Turk)
  - Real users

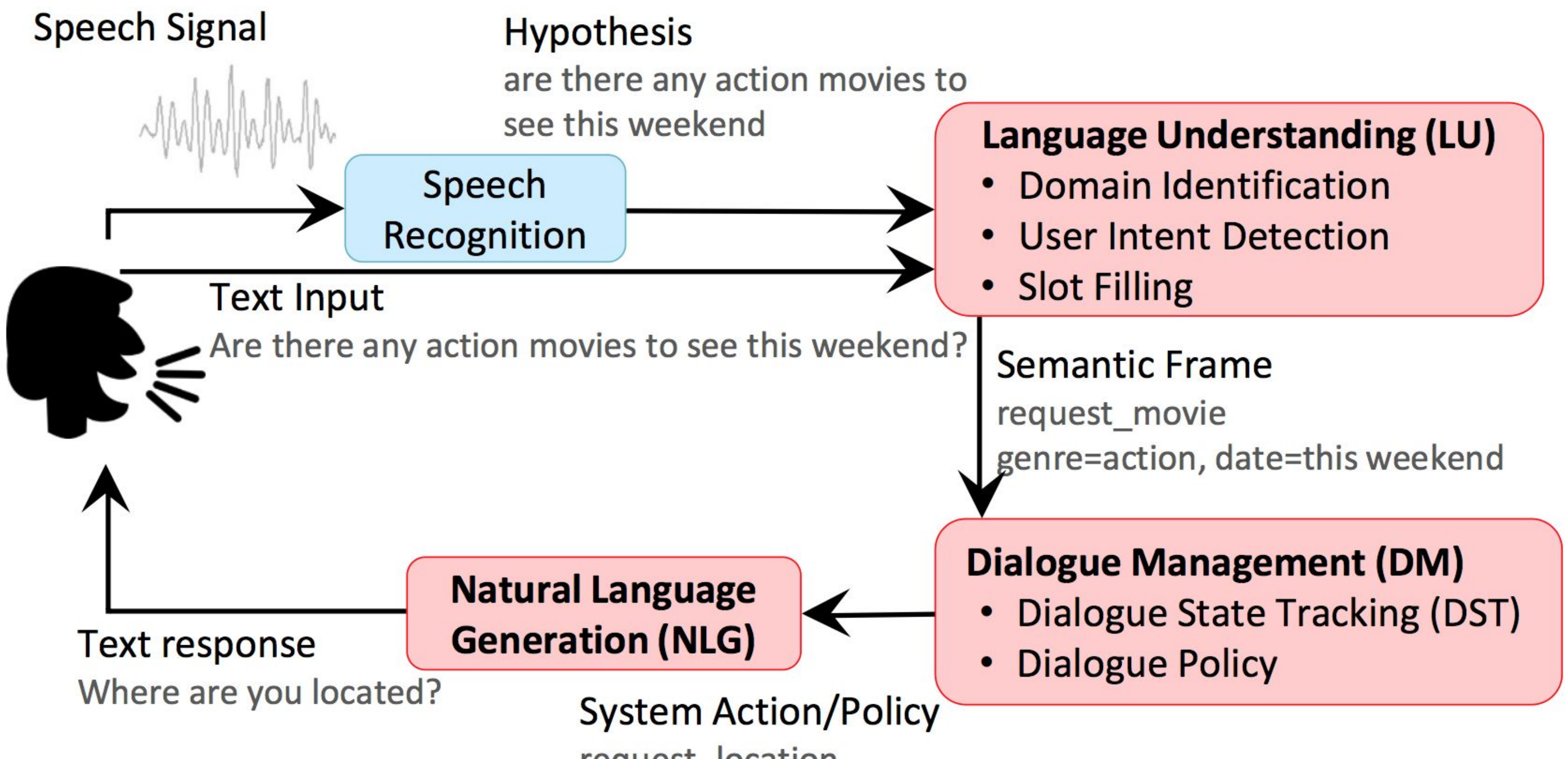


User Simulation

- Goal: generate natural and reasonable conversations to enable reinforcement learning for exploring the policy space
- Approach
  - Rule-based crafted by experts (Li et al., 2016)
  - Learning-based (Schatzmann et al., 2006)



Task-Oriented Dialogue System



Natural Language Generation

Mapping semantic frame into natural language

inform(name=Seven_Days, foodtype=Chinese)

vvv

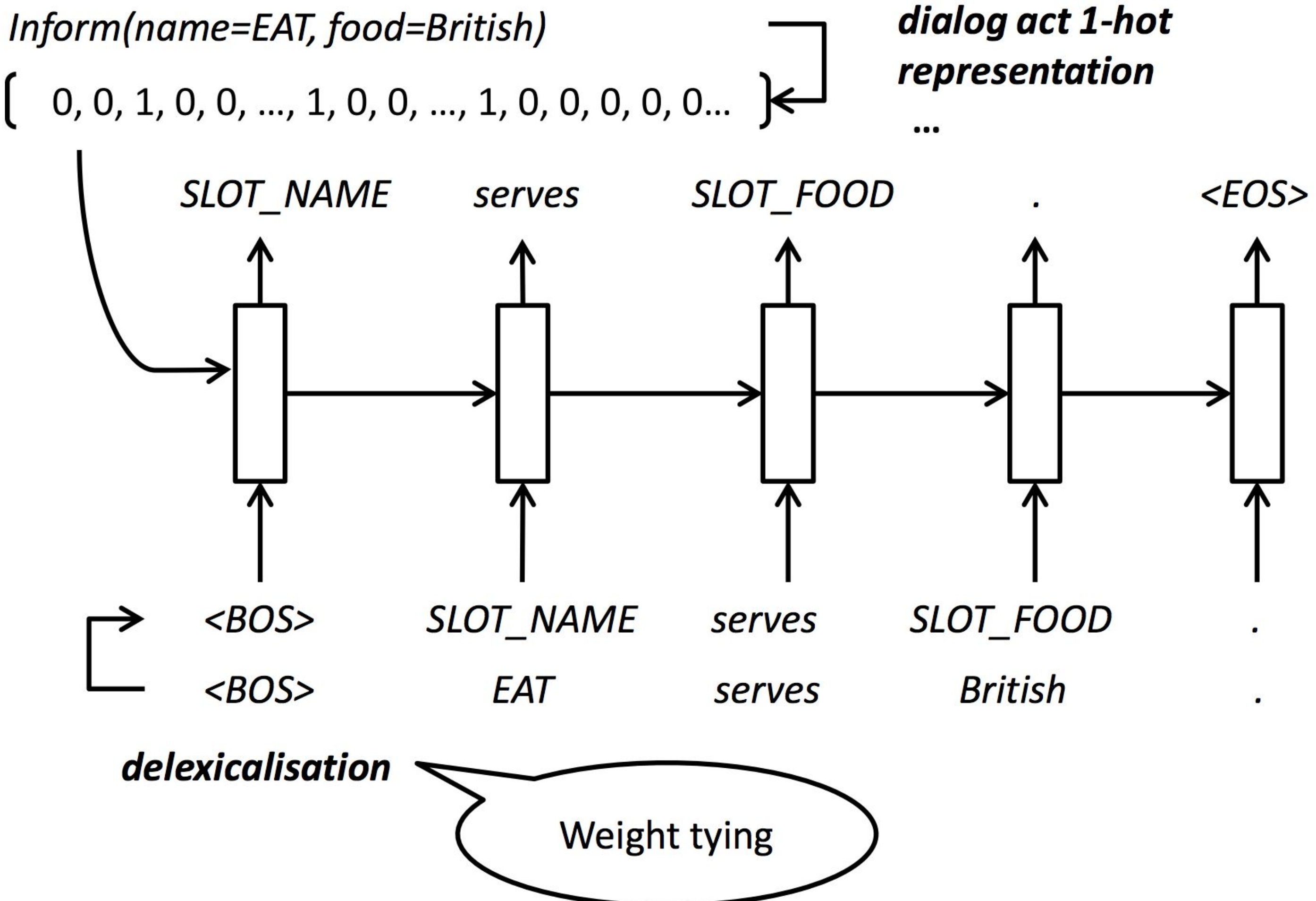
Seven Days is a nice Chinese restaurant

Template-Based Generator

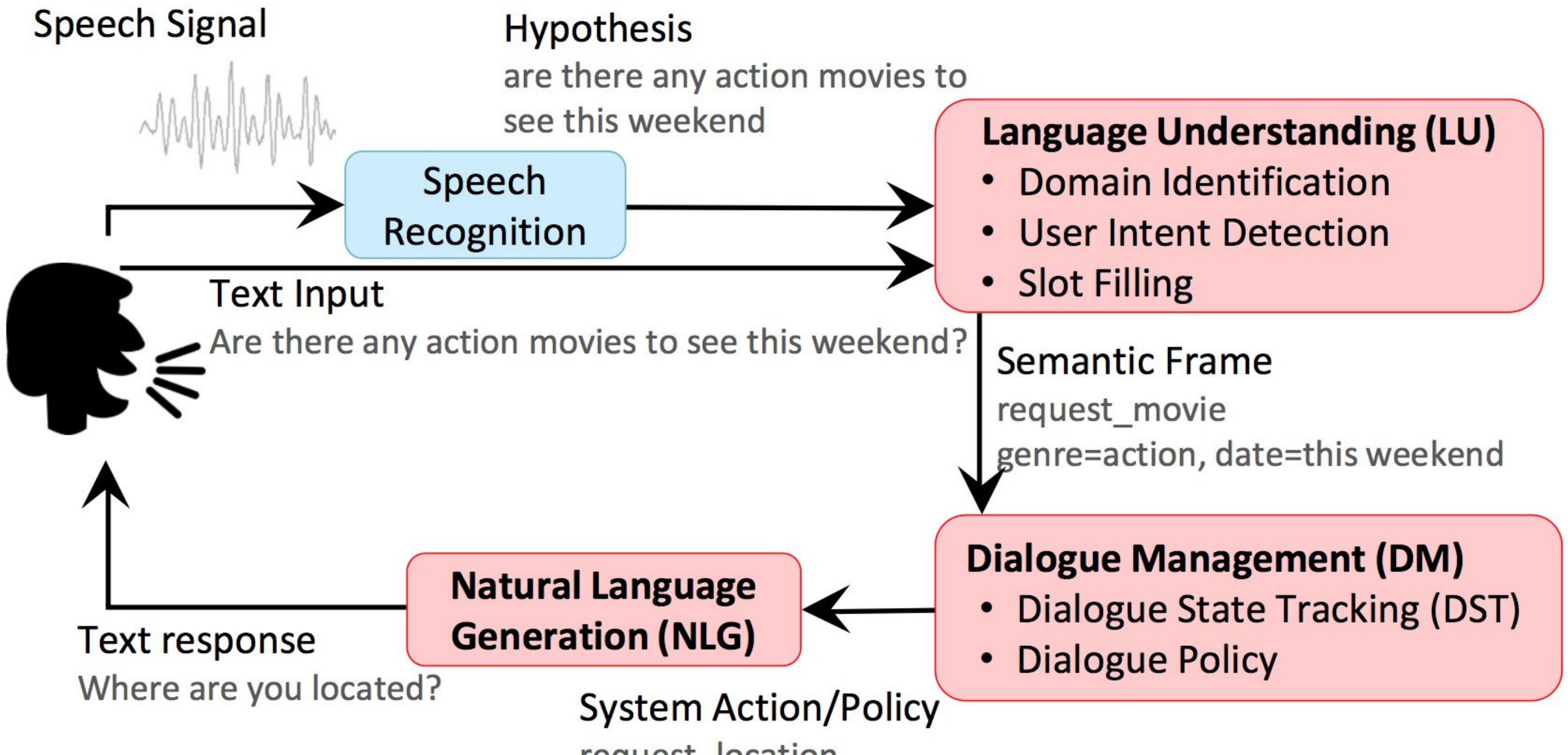
Semantic Frame	Natural Language
confirm()	"Please tell me more about the product your are looking for."
confirm(area=$V)	"Do you want somewhere in the $V?"
confirm(food=$V)	"Do you want a $V restaurant?"
confirm(food=$V,area=$W)	"Do you want a $V restaurant in the $W."

Pros: simple, error-free, easy to control
Cons: time-consuming, poor scalability

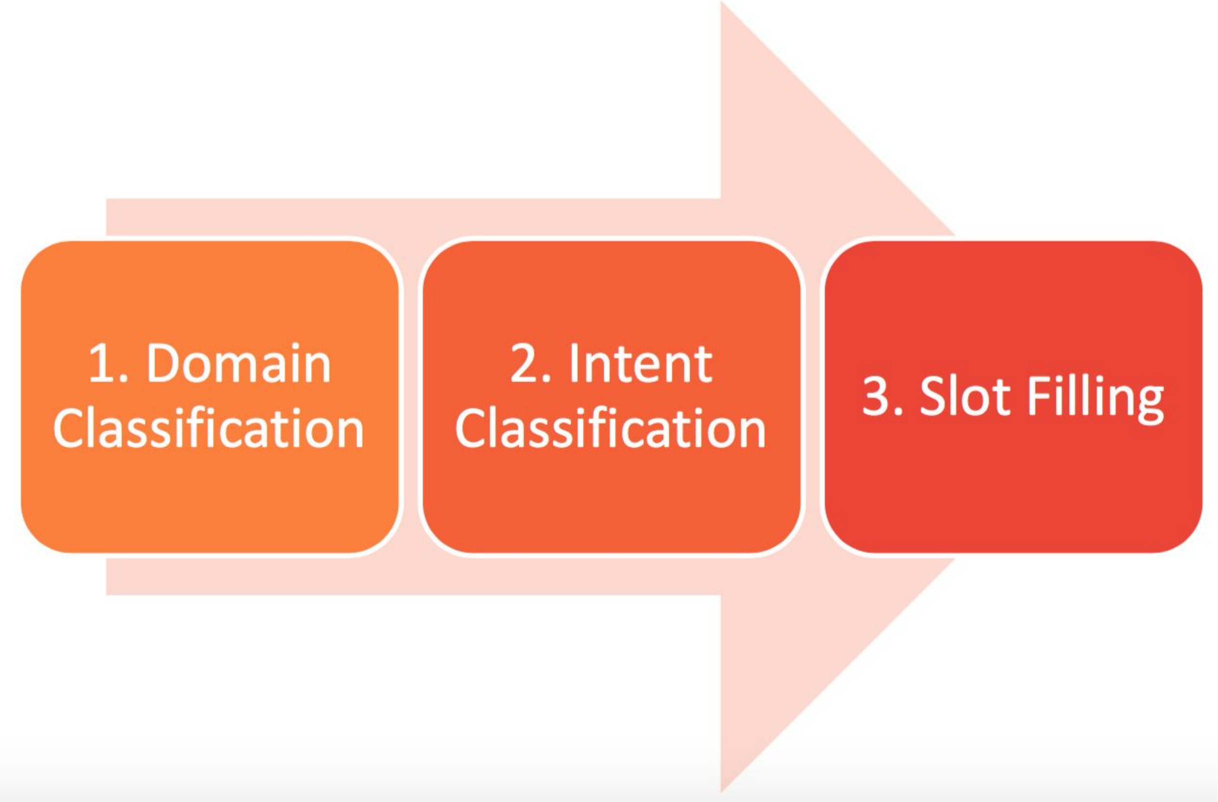
RNN Based Generator



Task-Oriented Dialogue System



NLU Pipeline

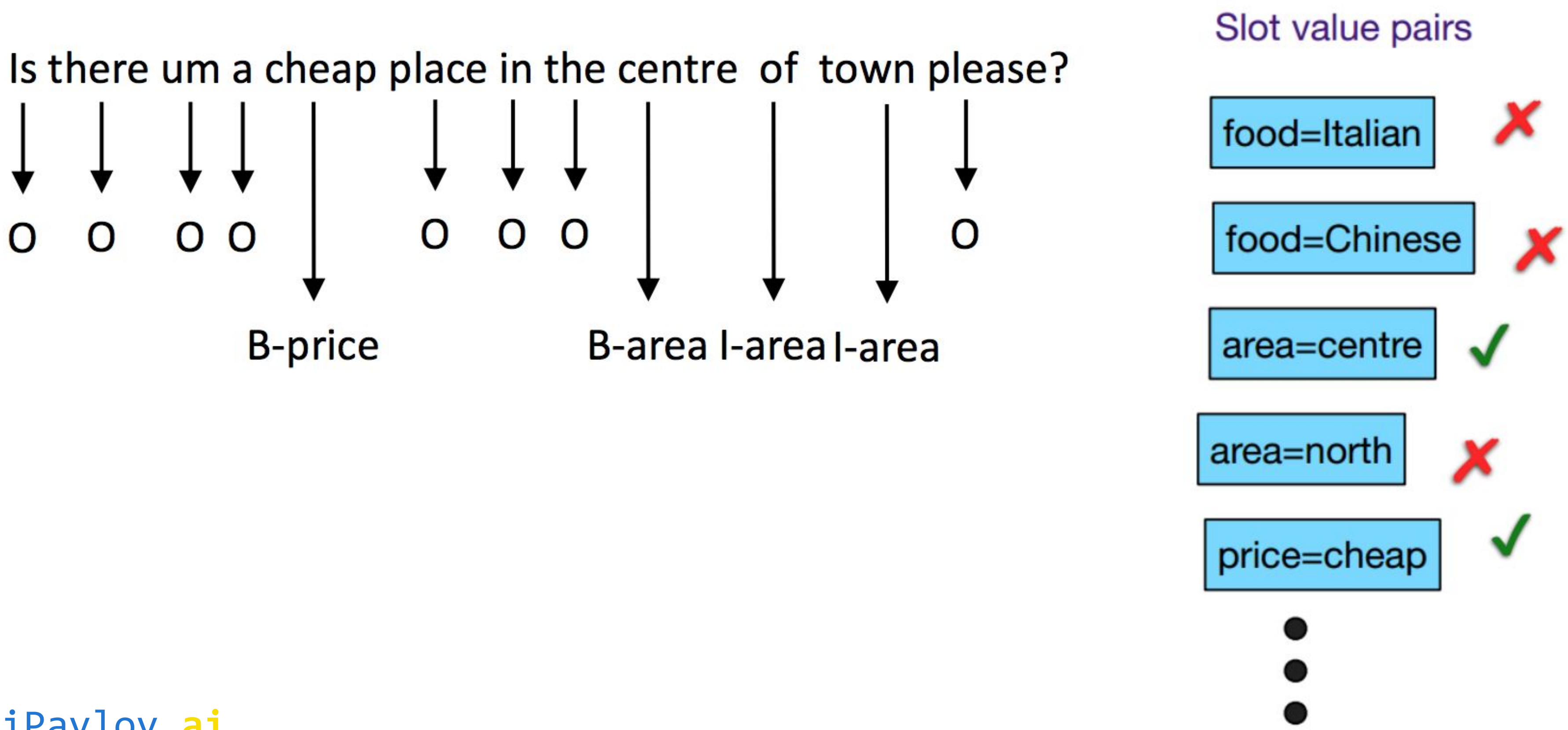


Domain/Task Classification

find me a cheap taiwanese restaurant in oakland

Movies	Find_movie
Restaurants	Buy_tickets
Sports	Find_restaurant
Weather	Book_table
Music	Find_lyrics
…	…

Slot Filling



Conventional Approach

Data — dialogue utterances annotated with domains/intents

Model — machine learning classification model e.g. support vector machine (SVM)

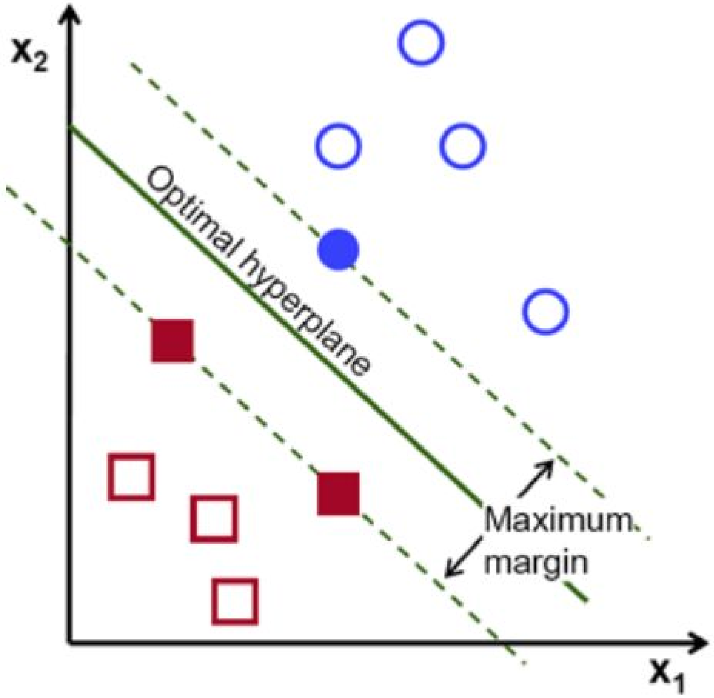
Prediction — domains/intents

iP

Theory: Support Vector Machine

SVM is a maximum margin classifier

- Input data points are mapped into a high dimensional feature space where the data is linearly separable
- Support vectors are input data points that lie on the margin



Slot Filling - Sequence Tagging

flights from Boston to New York today

	flights	from	Boston	to	New	York	today
Entity Tag	O	O	B-city	O	B-city	I-city	O
Slot Tag	O	O	B-dept	O	B-arrival	I-arrival	B-date

Conventional Approach

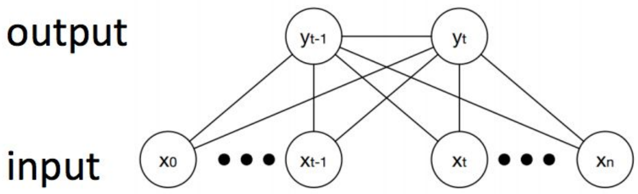
Data — dialogue utterances annotated with slots

Model — machine learning tagging model
e.g. conditional random fields (CRF)

Prediction — slots and their values

Theory: Conditional Random Fields

CRF assumes that the label at time step t depends on the label in the previous time step t-1



Maximize the log probability log p(y | x) with respect to parameters λ

$$p(y \mid x) = \frac{1}{Z(x)} \exp(\sum_i \lambda_i f_i(x, y))$$

$$= \prod_t \frac{1}{Z(x)} \exp(\sum_i \lambda_i f_i(x, y_t, y_{t-1}))$$

Slots can be tagged based on the y that maximizes $p(y|x)$

Deep Learning Approach

Data — dialogue utterances annotated with semantic frames (user intents & slots)

Model — deep learning model (classification/tagging)
e.g. recurrent neural networks (RNN)

Prediction — user intents, slots and their values

Word Representation

The vast majority of rule-based and sta4s4cal NLP work regards words as atomic symbols: hotel, conference, walk

In vector space terms, this is a vector with one 1 and a lot of zeroes

Dimensionality: 20K (speech) – 50K (PTB) – 500K (big vocab) – 13M (Google 1T)

We call this a “one-hot” representation. Its problem:

motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0

Distributional similarity based representations

- You can get a lot of value by representing a word by means of its neighbors

"You shall know a word by the company it keeps"

(J. R. Firth 1957: 11)

- One of the most successful ideas of modern statistical NLP

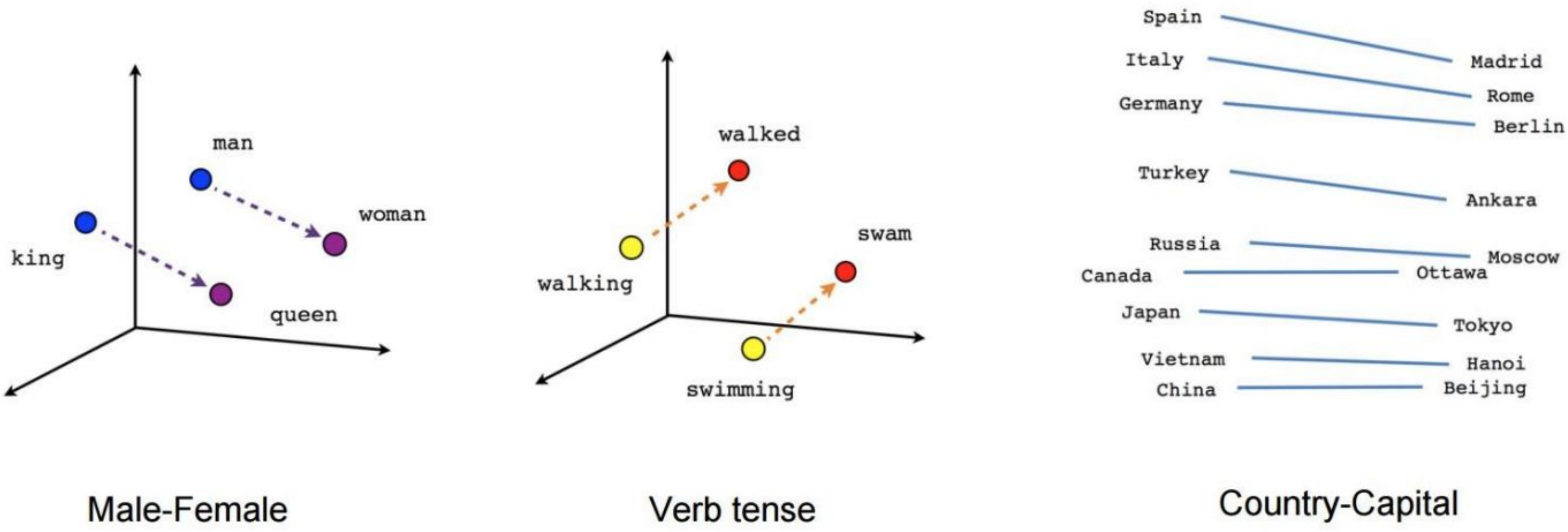
Distributional similarity based representations

government debt problems turning into banking crises as has happened in

saying that Europe needs unified banking regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

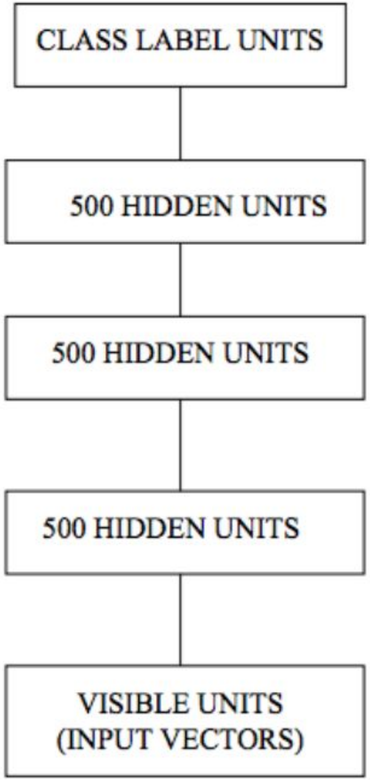
Distributional similarity based representations



Deep Neural Networks for Domain/Intent Classification – I (Sarikaya et al, 2011)

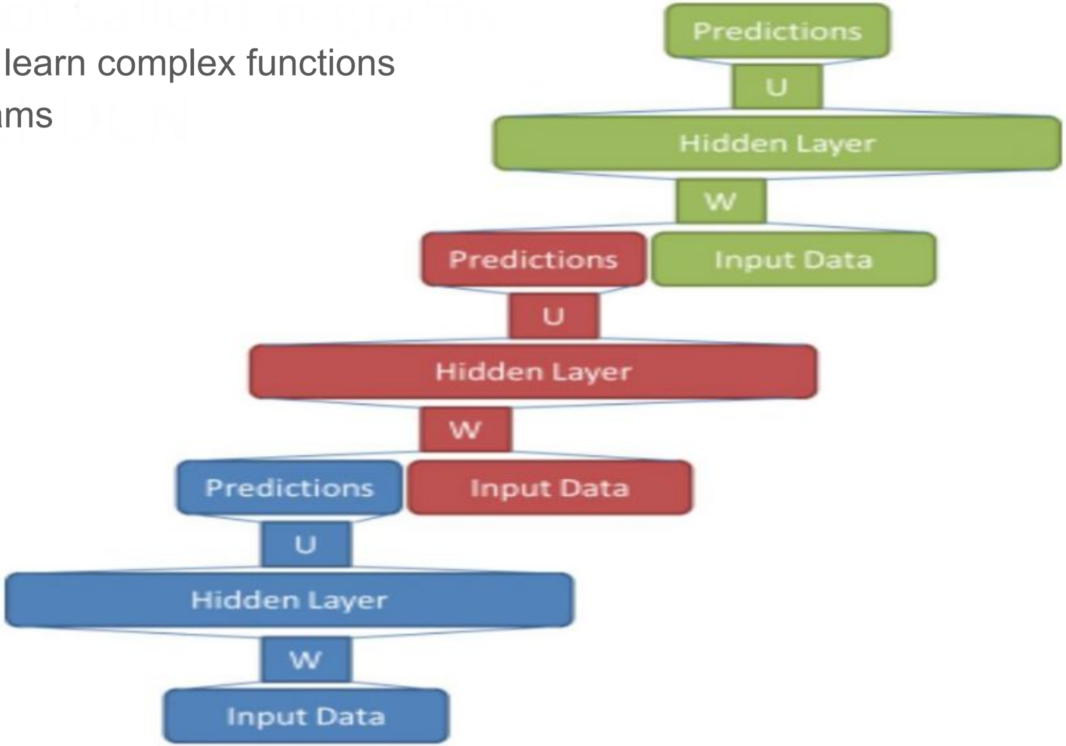
Deep belief nets (DBN)

- Unsupervised training of weights
- Fine-tuning by back-propagation
- Compared to MaxEnt, SVM, and boosting



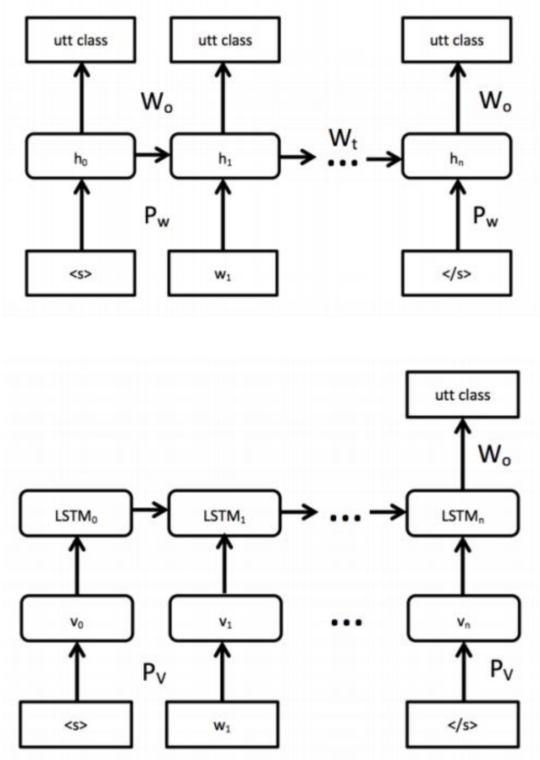
DNNs for Domain/Intent Classification – II (Tur et al., 2012; Deng et al., 2012)

- Deep convex networks (DCN)
  - Simple classifiers are stacked to learn complex functions
  - Feature selection of salient n-grams
- Extension to kernel-DCN



DNNs for Domain/Intent Classification – III (Ravuri and Stolcke, 2015)

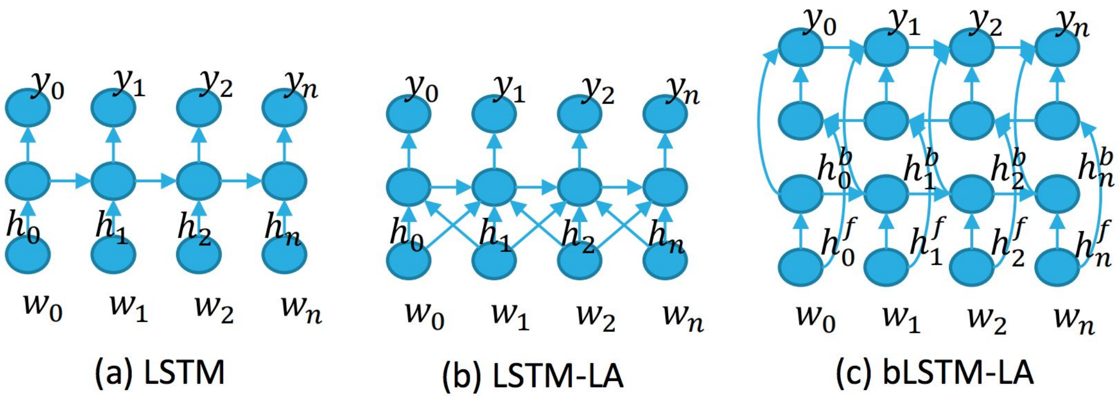
- RNN and LSTMs for utterance classification
- Word hashing to deal with large number of singletons
  - Kat: #Ka, Kat, at#
  - Each character n-gram is associated with a bit in the input encoding



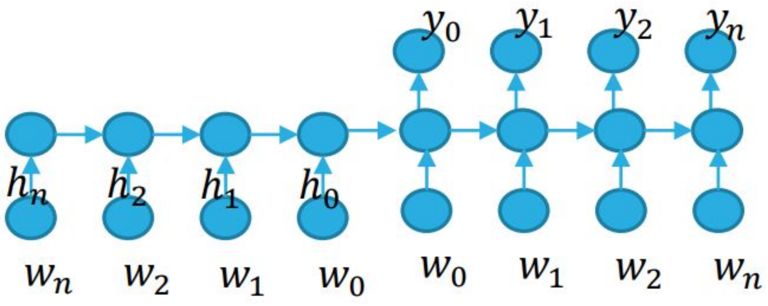
Recurrent Neural Nets for Slot Tagging – I (Yao et al, 2013; Mesnil et al, 2015)

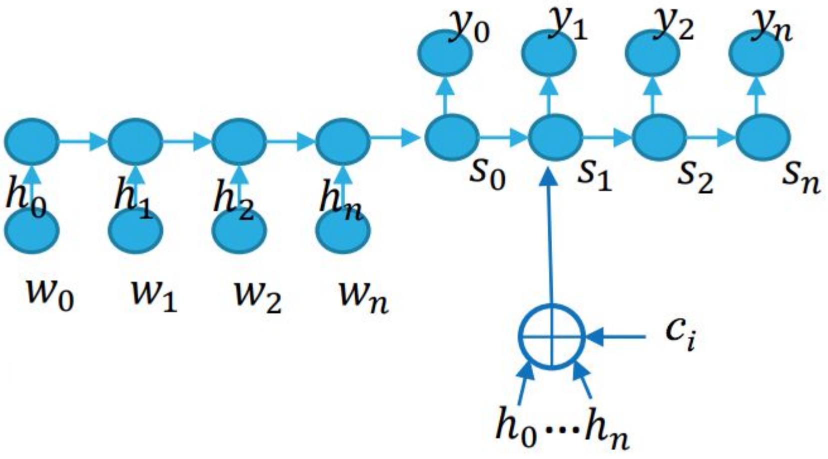
Variations:

A. RNNs with LSTM cells
B. Input, sliding window of n-grams
C. Bi-directional LSTMs



- Encoder-decoder networks
  - Leverages sentence level information
- Attention-based encoder-decoder
  - Use of attention (as in MT) in the encoder-decoder network
  - Attention is estimated using a feed-forward network with input: h and s at time t

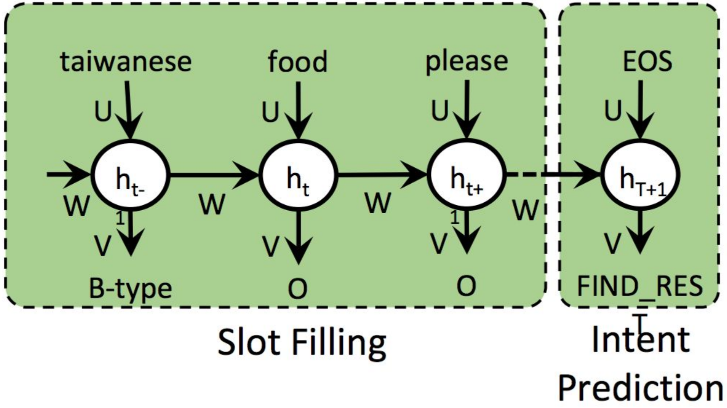


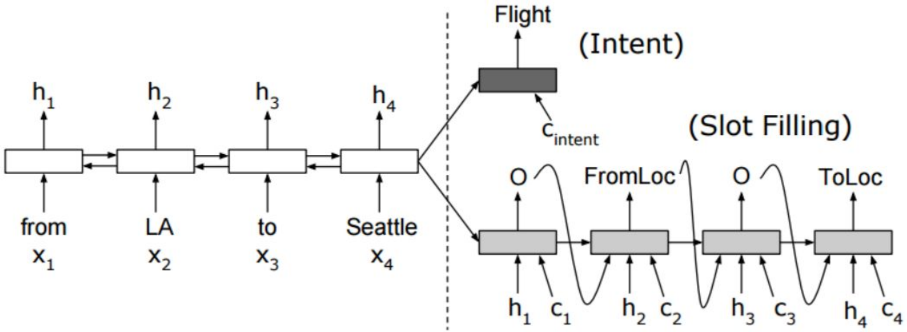


Joint Semantic Frame Parsing

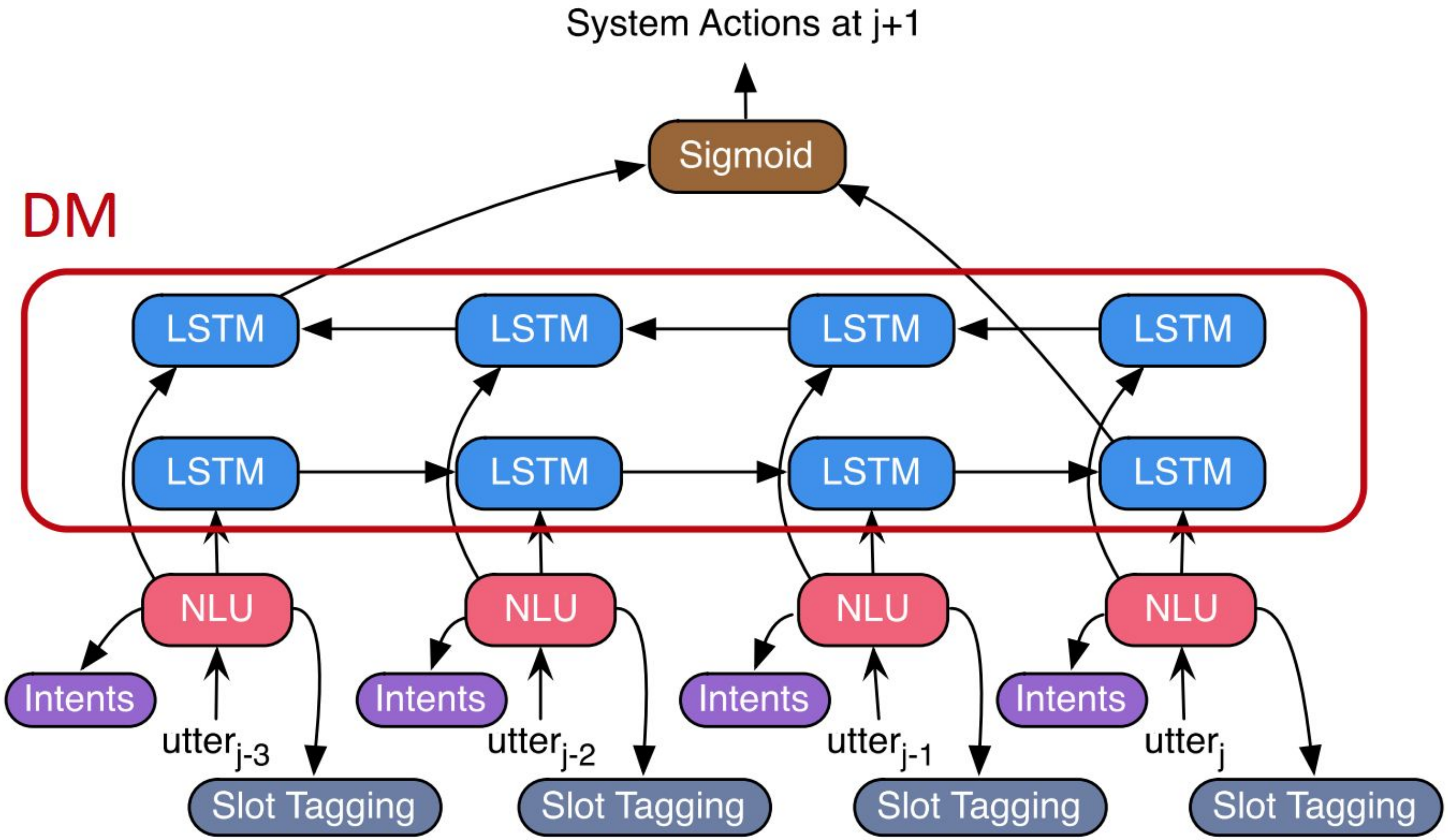
Slot filling and intent prediction in the same output sequence

Intent prediction and slot filling are performed in two “heads”



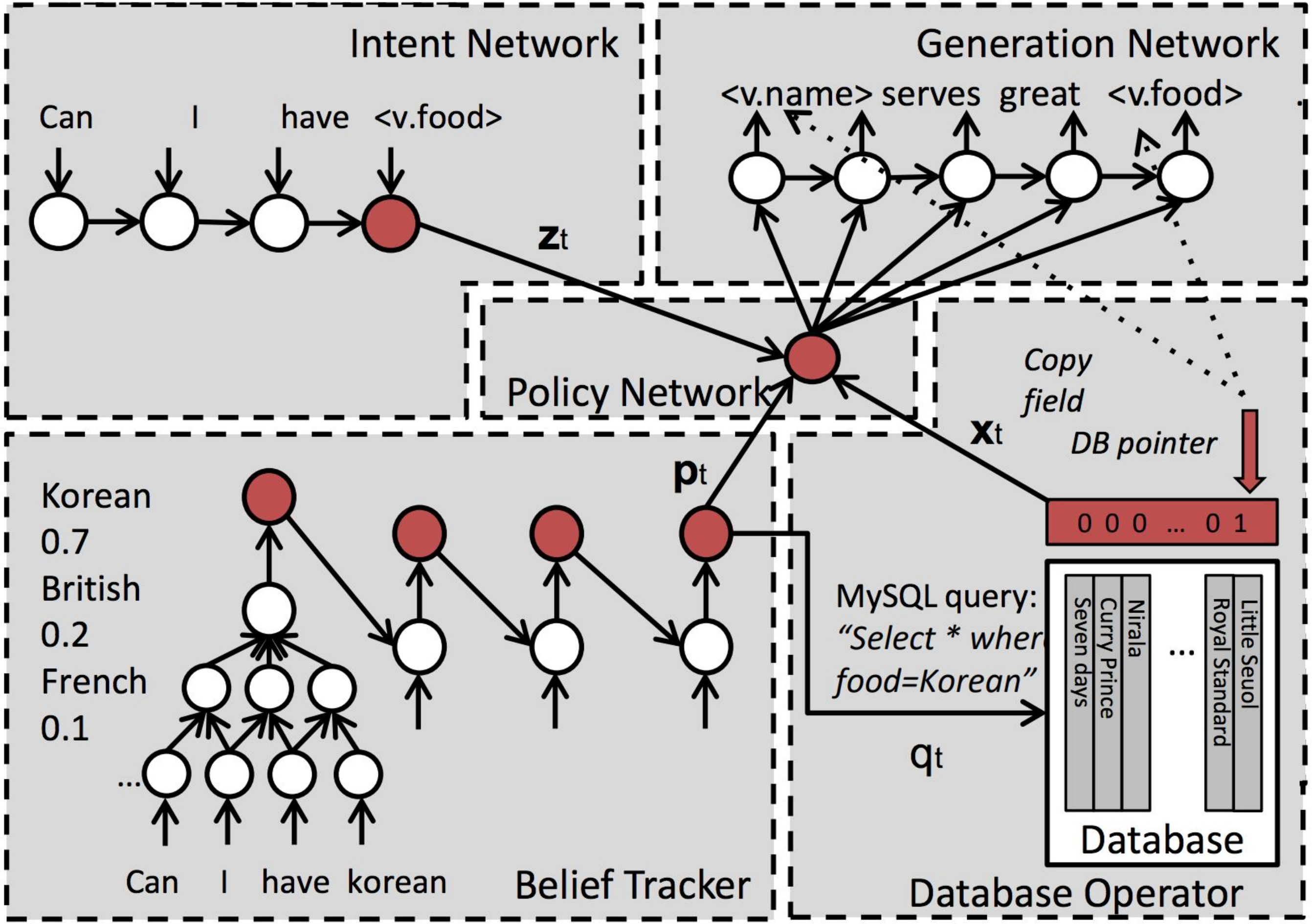


End-to-End



System Actions at j+1
Sigmoid
DM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
NLU
NLU
NLU
NLU
Intents
Intents
Intents
Intents
utter$_{j-3}$
utter$_{j-2}$
utter$_{j-1}$
utter$_{j}$
Slot Tagging
Slot Tagging
Slot Tagging
Slot Tagging

E2E Dialog System - Wen et al. 2016



Language Understanding

- Collect and annotate data
- Use machine learning method to train your system
- Conventional
  - SVM for domain/intent classification
  - CRF for slot filling
- Deep learning
  - LSTM for domain/intent classification and slot filling
- Test your system performance

Questions?

Acknowledgements

I would like to gratefully thank Vivian Chen from Taiwan National University for permission to use her materials to create this presentation.

Also I would like to thank Richard Socher for the slide about word embeddings.

Most of architecture pictures are belongs to authors of the papers mentioned. If you do not see any attribution on the picture, most probably I’ve missed the reference, write me and I add it.

Intent Classification / Slot Filling

Intent Classification (Ravuri and Stolcke, 2015)

IOB Sequence Labeling for Slot Filling (Hakkani-Tur et al., 2016)