WE UNDERSTAND
YOUR NEEDS

Intelligent Conversational Bot

YUN-NUNG (VIVIAN) CHEN

WWW.CSIE.NTU.EDU.TW/~YVCHEN/S105-ICB

Language Understanding
Mar 21st, 2017
Review
Task-Oriented Dialogue System (Young, 2000)

Speech Signal

Hypothesis
are there any action movies to see this weekend

Speech Recognition

Text Input
Are there any action movies to see this weekend?

Semantic Frame
request_movie
genre=action, date=this weekend

Language Understanding (LU)
- Domain Identification
- User Intent Detection
- Slot Filling

Dialogue Management (DM)
- Dialogue State Tracking (DST)
- Dialogue Policy

System Action/Policy
request_location

Text response
Where are you located?

Backend Database/Knowledge Providers

Natural Language Generation (NLG)
Task-Oriented Dialogue System (Young, 2000)

Speech Signal

Hypothesis
are there any action movies to see this weekend

Language Understanding (LU)
• Domain Identification
• User Intent Detection
• Slot Filling

Semantic Frame
request_movie
genre=action, date=this weekend

Dialogue Management (DM)
• Dialogue State Tracking (DST)
• Dialogue Policy

Backend Action / Knowledge Providers

Text Input
Are there any action movies to see this weekend?

Speech Recognition

Text response
Where are you located?

Natural Language Generation (NLG)

System Action/Policy
request_location
Conventional LU
Language Understanding (LU)

- Pipelined

1. Domain Classification
2. Intent Classification
3. Slot Filling
LU – Domain/Intent Classification

As an utterance classification task

• Given a collection of utterances \( u_i \) with labels \( c_i \),
  \( D = \{(u_1,c_1),..., (u_n,c_n)\} \) where \( c_i \in C \), train a model to estimate labels for new utterances \( u_k \).

find me a cheap taiwanese restaurant in oakland

Movies   find_movie, buy_tickets
Restaurants find_restaurant, find_price, book_table
Music     find_lyrics, find_singer
Sports    ...
...

Domain    Intent
Conventional Approach

Data
- dialogue utterances annotated with domains/intents

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

Prediction
- domains/intents
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

![Diagram of SVM with optimal hyperplane and maximum margin](http://www.csie.ntu.edu.tw/~htlin/mooc/)
Theory: Support Vector Machine

- Multiclass SVM
  - Extended using **one-versus-rest** approach
  - Then transform into probability

Domain/intent can be decided based on the estimated scores
LU – Slot Filling

As a sequence tagging task

• Given a collection tagged word sequences, 
  \[ S = \{((w_{1,1}, w_{1,2}, \ldots, w_{1,n_1}), (t_{1,1}, t_{1,2}, \ldots, t_{1,n_1})),
           ((w_{2,1}, w_{2,2}, \ldots, w_{2,n_2}), (t_{2,1}, t_{2,2}, \ldots, t_{2,n_2})),
           \ldots\} \]
  where \( t_i \in M \), the goal is to estimate tags for a new word sequence.

flights from Boston to New York today

<table>
<thead>
<tr>
<th>Entity Tag</th>
<th>Slot Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>flights</td>
<td>from</td>
</tr>
<tr>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>
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 $t-1$

```
input: x0 • • • xt-1
output: yt-1 yt
```

- Maximize the log probability $\log p(y \mid x)$ with respect to parameters $\lambda$

\[
p(y \mid x) = \frac{1}{Z(x)} \exp\left( \sum_i \lambda_i f_i(x, y) \right) = \prod_t \frac{1}{Z(x)} \exp\left( \sum_i \lambda_i f_i(x, y_t, y_{t-1}) \right)
\]

Slots can be tagged based on the $y$ that maximizes $p(y \mid x)$
Neural Network Based LU
A Single Neuron

\[ y = h_{w,b}(x) = \sigma(w^T x + b) \]

**Activation function**

\[ \sigma(z) = \frac{1}{1 + e^{-z}} \]

**Sigmoid function**

\[ w, b \text{ are the parameters of this neuron} \]
A single neuron can only handle binary classification.
A Layer of Neurons

- Handwriting digit classification

A layer of neurons can handle multiple possible output, and the result depends on the max one.

$f : R^N \rightarrow R^M$

10 neurons/10 classes

Which one is max?
Deep Neural Networks (DNN)

- Fully connected feedforward network

$$f : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

Deep NN: multiple hidden layers
Recurrent Neural Network (RNN)

\[ s_t = \sigma(Ws_{t-1} + Ux_t) \quad \sigma(\cdot): \text{tanh, ReLU} \]

\[ o_t = \text{softmax}(Vs_t) \]

RNN can learn accumulated sequential information (time-series)

Model Training

- All model parameters \( \theta = \{U, V, W\} \) can be updated by SGD.
The model is trained by comparing the correct sequence tags and the predicted ones.
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
Classification Model

- Given a collection of utterances $u_i$ with labels $c_i$, $D = \{(u_1, c_1), \ldots, (u_n, c_n)\}$ where $c_i \in C$, train a model to estimate labels for new utterances $u_k$.

- Input: each utterance $u_i$ is represented as a feature vector $f_i$
- Output: a domain/intent label $c_i$ for each input utterance

How to represent a sentence using a feature vector
Sequence Tagging Model

- Given a collection tagged word sequences, \( S = \{(w_{1,1}, w_{1,2}, \ldots, w_{1,n_1}), (t_{1,1}, t_{1,2}, \ldots, t_{1,n_1}), (w_{2,1}, w_{2,2}, \ldots, w_{2,n_2}), (t_{2,1}, t_{2,2}, \ldots, t_{2,n_2}), \ldots\} \)
  where \( t_i \in M \), the goal is to estimate tags for a new word sequence.

- Input: each word \( w_{i,j} \) is represented as a feature vector \( f_{i,j} \)
- Output: a slot label \( t_i \) for each word in the utterance

How to represent a word using a feature vector
Word Representation

- Atomic symbols: *one-hot* representation

\[
\text{car} \quad [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ \ldots \ 0]
\]

Issues: difficult to compute the similarity (i.e. comparing “car” and “motorcycle”)

\[
[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ \ldots \ 0] \quad \text{AND} \quad [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \ldots \ 0] = 0
\]
Word Representation

- Neighbor-based: low-dimensional dense word embedding

Idea: words with similar meanings often have similar neighbors
Chinese Input Unit of Representation

- Character
  - Feed each char to each time step

- Word
  - Word segmentation required

Can two types of information fuse together for better performance?
As an utterance classification task, given a collection of utterances $u_i$ with labels $c_i$, $D = \{(u_1, c_1), ..., (u_n, c_n)\}$ where $c_i \in C$, train a model to estimate labels for new utterances $u_k$.

Example utterance: find me a cheap Taiwanese restaurant in Oakland.

- **Domain Intent**
  - Movies: find_movie, buy_tickets
  - Restaurants: find_restaurant, find_price, book_table
  - Music: find_lyrics, find_singer
  - Sports: ...
  - ...
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
Deep Neural Networks 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

Deep Neural Networks 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

LU – Slot Filling

As a sequence tagging task

Given a collection tagged word sequences,
\[ S = \{ ((w_{1,1}, w_{1,2}, \ldots, w_{1,n1}), (t_{1,1}, t_{1,2}, \ldots, t_{1,n1})), ((w_{2,1}, w_{2,2}, \ldots, w_{2,n2}), (t_{2,1}, t_{2,2}, \ldots, t_{2,n2})), \ldots \} \]
where \( t_i \in M \), the goal is to estimate tags for a new word sequence.

flights from Boston to New York today

<table>
<thead>
<tr>
<th>Entity Tag</th>
<th>Slot Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>flights</td>
<td>from</td>
</tr>
<tr>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>
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

![Diagram](http://131.107.65.14/en-us/um/people/gzweig/Pubs/Interspeech2013RNNLU.pdf; http://dl.acm.org/citation.cfm?id=2876380)
Recurrent Neural Nets for Slot Tagging – II
(Kurata et al., 2016; Simonnet et al., 2015)

- 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: $h_t$ and $s_t$ at time $t$
Joint Semantic Frame Parsing

• Slot filling and intent prediction in the same output sequence

Sequence-based (Hakkani-Tur et al., 2016)

Parallel (Liu and Lane, 2016)

• Intent prediction and slot filling are performed in two branches

Milestone 1 – Language Understanding

3) Collect and annotate data
4) 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
5) Test your system performance
Concluding Remarks

Speech Signal
Hypothesis
are there any action movies to see this weekend

Text Input
Are there any action movies to see this weekend?

Speech Recognition

Language Understanding (LU)
- Domain Identification
- User Intent Detection
- Slot Filling

Semantic Frame
request_movie
genre=action, date=this weekend

Natural Language Generation (NLG)

System Action/Policy
request_location

Dialogue Management (DM)
- Dialogue State Tracking (DST)
- Dialogue Policy

Backend Database/
Knowledge Providers

Text response
Where are you located?