Unsupervised Learning and Modeling of Knowledge and Intent for Spoken Dialogue Systems

Yun-Nung (Vivian) Chen
Email yvchen@cs.cmu.edu
Website http://vivianchen.idv.tw
OUTLINE

Introduction

Ontology Induction: Frame-Semantic Parsing

Structure Learning: Knowledge Graph Propagation

Spoken Language Understanding (SLU): Matrix Factorization

Experiments

Conclusions
OUTLINE

Introduction
- Ontology Induction: Frame-Semantic Parsing
- Structure Learning: Knowledge Graph Propagation
- Spoken Language Understanding (SLU): Matrix Factorization
- Experiments
- Conclusions
A POPULAR ROBOT - BAYMAX

Baymax is capable of maintaining a good spoken dialogue system and learning new knowledge for better understanding and interacting with people.
SPOKEN DIALOGUE SYSTEM (SDS)

Spoken dialogue systems are the intelligent agents that are able to help users finish tasks more efficiently via speech interactions.

Spoken dialogue systems are being incorporated into various devices (smart-phones, smart TVs, in-car navigating system, etc).

Apple’s Siri
Microsoft’s Cortana
Microsoft’s XBOX Kinect
Amazon’s Echo
Samsung’s SMART TV
Google Now

https://www.apple.com/ios/siri/
http://www.amazon.com/oc/echo/
https://www.google.com/landing/now/
LARGE SMART DEVICE POPULATION

The number of global smartphone users will surpass 2 billion in 2016.

As of 2012, there are 1.1 billion automobiles on the earth.

The more **natural** and **convenient** input of the devices evolves towards **speech**.
CHALLENGES FOR SDS

An SDS in a new domain requires
1) A hand-crafted domain ontology
2) Utterances labeled with semantic representations
3) An SLU component for mapping utterances into semantic representations

With increasing spoken interactions, building domain ontologies and annotating utterances cost a lot so that the data does not scale up.

The goal is to enable an SDS to automatically learn this knowledge so that open domain requests can be handled.
**INTERACTION EXAMPLE**

User: find an inexpensive eating place for Taiwanese food

Intelligent Agent: Inexpensive Taiwanese eating places include Din Tai Fung, etc. What do you want to choose? I can help you go there.

Q: How does a dialogue system process this request?
SDS PROCESS – AVAILABLE DOMAIN ONTOLOGY

User

find an inexpensive eating place for taiwanese food

Intelligent Agent

Organized Domain Knowledge

price

AMOD

food

NN

target

PREP_FOR

seeking

food
find an inexpensive eating place for Taiwanese food

**Ontology Induction** *(semantic slot)*

- **target**
  - **price** (AMOD)
  - **food** (NN)
  - **seeking** (PREP_FOR)

**Organized Domain Knowledge**
User

find an inexpensive eating place for taiwanese food

Ontology Induction *(semantic slot)*

price

AMOD

target

food

NN

seeking

PREP_FOR

Structure Learning *(inter-slot relation)*

Organized Domain Knowledge
The SDS PROCESS – Spoken Language Understanding (SLU) involves finding an inexpensive eating place for Taiwanese food. The organized domain knowledge for this task includes:

- **Seeking**: "find"
- **Target**: "eating place"
- **Price**: "inexpensive"
- **Food**: "Taiwanese food"
User

**find** an inexpensive eating place for **taiwanese food**

SELECT restaurant {
    restaurant.price="inexpensive"
    restaurant.food="taiwanese food"
}

**Predicted behavior:** navigation

**Behavior Prediction**

Intelligent Agent

Inexpensive Taiwanese eating places include Din Tai Fung, Boiling Point, etc. What do you want to choose? I can help you go there. (navigation)
GOALS

1. Ontology Induction (semantic slot)

2. Structure Learning (inter-slot relation)

3. Spoken Language Understanding

SELECT restaurant {
  restaurant.price=“inexpensive”
  restaurant.food=“taiwanese food”
}

Predicted behavior: navigation

4. Behavior Prediction

User

find an inexpensive eating place for taiwanese food
find an inexpensive eating place for taiwanese food

1. Ontology Induction
2. Structure Learning
3. Semantic Decoding
4. Behavior Prediction

GOALS
SPOKEN LANGUAGE UNDERSTANDING

Input: user utterances

Output: the domain-specific semantic concepts included in each utterance
OUTLINE

Introduction

Ontology Induction: Frame-Semantic Parsing

Structure Learning: Knowledge Graph Propagation

Spoken Language Understanding (SLU): Matrix Factorization

Experiments

Conclusions
PROBABILISTIC FRAME-SEMANTIC PARSEING

FrameNet [Baker et al., 1998]
- a linguistically semantic resource, based on the frame-semantics theory
- words/phrases can be represented as frames
- “low fat milk” → “milk” evokes the “food” frame;
  “low fat” fills the descriptor frame element

SEMAFOR [Das et al., 2014]
- a state-of-the-art frame-semantics parser, trained on manually annotated FrameNet sentences

can i have a cheap restaurant

Frame: capability
FT LU: can FE LU: i

Frame: expensiveness
FT LU: cheap

Frame: locale by use
FT/FE LU: restaurant

Good!

1st Issue: adapting generic frames to domain-specific settings for SDSs
SPOKEN LANGUAGE UNDERSTANDING

Input: user utterances

Output: the domain-specific semantic concepts included in each utterance

SLU Modeling by Matrix Factorization

- Ontology Induction
  - Frame-Semantic Parsing
- Word Relation Model
  - Structure Learning
  - Slot Relation Model
  - Lexical KG
  - Semantic KG

- Feature Model
  - $F_w$
  - $F_s$

- Knowledge Graph Propagation Model
  - $R_w$
  - $R_s$

- SLU Model

"can I have a cheap restaurant"

target="restaurant"
price="cheap"

Semantic Representation

OUTLINE

Introduction

Ontology Induction: Frame-Semantic Parsing

Structure Learning: Knowledge Graph Propagation
(for 1st issue)

Spoken Language Understanding (SLU): Matrix Factorization

Experiments

Conclusions
**1ST ISSUE:** HOW TO ADAPT GENERIC SLOTS TO DOMAIN-SPECIFIC SETTING?

**KNOWLEDGE GRAPH PROPAGATION MODEL**

Assumption: The domain-specific words/slots have more dependency to each other.

Relation matrices allow each node to propagate scores to its neighbors in the knowledge graph, so that domain-specific words/slots have higher scores after matrix multiplication.
KNOWLEDGE GRAPH CONSTRUCTION

Syntactic dependency parsing on utterances

**can**  **i**  **have**  **a**  **cheap**  **restaurant**

- capability
- expensiveness
- locale_by_use

Slot-based semantic knowledge graph

Word-based lexical knowledge graph
The edge between a node pair is weighted as relation importance to propagate the scores via a relation matrix.

How to decide the weights to represent relation importance?
WEIGHT MEASUREMENT BY EMBEDDINGS

Dependency-based word embeddings

can i have a cheap restaurant

can = [0.8  ...  0.24]

have = [0.3  ...  0.21]

Dependency-based slot embeddings

capability have a expensiveness locale_by_use

capability = [0.12  ...  0.7]

expensiveness = [0.3  ...  0.6]

WEIGHT MEASUREMENT BY EMBEDDINGS

Compute edge weights to represent relation importance

- Slot-to-slot semantic relation $R^S_s$: similarity between slot embeddings
- Slot-to-slot dependency relation $R^D_s$: dependency score between slot embeddings
- Word-to-word semantic relation $R^S_w$: similarity between word embeddings
- Word-to-word dependency relation $R^D_w$: dependency score between word embeddings

\[ R^{SD}_s = R^S_s + R^D_s \]

\[ R^{SD}_w = R^S_w + R^D_w \]

KNOWLEDGE GRAPH PROPAGATION MODEL

Word Observation

cheap  food  restaurant

Slot Candidate

expensiveness  food  locale_by_use

Word Relation Model

Slot Relation Model

Slot Induction

$R_{SD}^S$

word relation matrix

$R_{SD}^w$

slot relation matrix

Train

Test
**FEATURE MODEL**

Utterance 1

*i would like a cheap restaurant*

Utterance 2

*find a restaurant with chinese food*

Test Utterance

*show me a list of cheap restaurants*

**2nd Issue:** unobserved hidden semantics may benefit understanding
OUTLINE

Introduction

Ontology Induction: Frame-Semantic Parsing

Structure Learning: Knowledge Graph Propagation

Spoken Language Understanding (SLU): Matrix Factorization

(for 2nd issue)

Experiments

Conclusions
MF method completes a partially-missing matrix based on a low-rank latent semantics assumption.
MATRIX FACTORIZATION (MF)

The decomposed matrices represent low-rank latent semantics for utterances and words/slots respectively.

The product of two matrices fills the probability of hidden semantics.
BAYESIAN PERSONALIZED RANKING FOR MF

Model implicit feedback

- not treat unobserved facts as negative samples (true or false)
- give observed facts higher scores than unobserved facts

\[ f^+ = \langle u, x^+ \rangle \quad p(f^+) > p(f^-) \]

\[ f^- = \langle u, x^- \rangle \]

\[ p(M_{u,x} = 1 \mid \theta_{u,x}) = \sigma(\theta_{u,x}) = \frac{1}{1 + \exp(-\theta_{u,x})} \]

Objective:

\[ \sum_{f^+ \in \mathcal{O}} \sum_{f^- \notin \mathcal{O}} \ln \sigma(\theta_{f^+} - \theta_{f^-}) \]

The objective is to learn a set of well-ranked semantic slots per utterance.
MF method completes a partially-missing matrix based on a low-rank latent semantics assumption.
EXPERIMENTAL SETUP

Dataset

- Cambridge University SLU corpus [Henderson, 2012]
  - Restaurant recommendation in an in-car setting in Cambridge
    - WER = 37%
    - vocabulary size = 1868
    - 2,166 dialogues
    - 15,453 utterances
    - dialogue slot: addr, area, food, name, phone, postcode, price range, task, type

EXPERIMENT 1: QUALITY OF SEMANTICS ESTIMATION

Metric: Mean Average Precision (MAP) of all estimated slot probabilities for each utterance

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o</td>
<td>w/ Explicit</td>
</tr>
<tr>
<td>Explicit</td>
<td>32.5</td>
<td>36.6</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multinomial Logistic Regression</td>
<td>34.0</td>
<td>38.8</td>
</tr>
</tbody>
</table>
EXPERIMENT 1: QUALITY OF SEMANTICS ESTIMATION

Metric: Mean Average Precision (MAP) of all estimated slot probabilities for each utterance

<table>
<thead>
<tr>
<th>Modeling Implicit Semantics</th>
<th>Approach</th>
<th>ASR w/o</th>
<th>ASR w/ Explicit</th>
<th>Manual w/o</th>
<th>Manual w/ Explicit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Explicit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support Vector Machine</td>
<td>32.5</td>
<td></td>
<td>36.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multinomial Logistic Regression</td>
<td>34.0</td>
<td></td>
<td>38.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implicit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Random</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Majority</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feature Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feature Model + Knowledge Graph Propagation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Experiment 1: Quality of Semantics Estimation

**Metric:** Mean Average Precision (MAP) of all estimated slot probabilities for each utterance

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o</td>
<td>w/ Explicit</td>
</tr>
<tr>
<td>Explicit Support Vector Machine</td>
<td>32.5</td>
<td></td>
</tr>
<tr>
<td>Multinomial Logistic Regression</td>
<td>34.0</td>
<td></td>
</tr>
<tr>
<td>Implicit Baseline Random</td>
<td>3.4</td>
<td></td>
</tr>
<tr>
<td>Majority</td>
<td>15.4</td>
<td></td>
</tr>
<tr>
<td>Implicit Feature Model</td>
<td>24.2</td>
<td></td>
</tr>
<tr>
<td>MF Feature Model + Knowledge</td>
<td>40.5*</td>
<td></td>
</tr>
<tr>
<td>Graph Propagation (+19.1%)</td>
<td>(+19.1%)</td>
<td></td>
</tr>
</tbody>
</table>
## EXPERIMENT 1: QUALITY OF SEMANTICS ESTIMATION

**Metric:** Mean Average Precision (MAP) of all estimated slot probabilities for each utterance

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o</td>
<td>w/ Explicit</td>
</tr>
<tr>
<td>Explicit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>32.5</td>
<td>36.6</td>
</tr>
<tr>
<td>Multinomial Logistic Regression</td>
<td>34.0</td>
<td>38.8</td>
</tr>
<tr>
<td>Implicit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>3.4</td>
<td>22.5</td>
</tr>
<tr>
<td>Majority</td>
<td>15.4</td>
<td>32.9</td>
</tr>
<tr>
<td>MF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature Model</td>
<td>24.2</td>
<td>37.6*</td>
</tr>
<tr>
<td>Feature Model + Knowledge Graph Propagation</td>
<td>40.5*</td>
<td>43.5*</td>
</tr>
</tbody>
</table>

* (+19.1%) ( +27.9%) ( +34.3%) ( +37.6%)

The MF approach effectively models hidden semantics to improve SLU.

Adding a knowledge graph propagation model further improves performance.
All types of relations are useful to infer hidden semantics.
## EXPERIMENT 2: EFFECTIVENESS OF RELATIONS

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Model</td>
<td>37.6</td>
<td>45.3</td>
</tr>
<tr>
<td>Feature + Knowledge Graph Propagation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic</td>
<td>41.4*</td>
<td>51.6*</td>
</tr>
<tr>
<td>Dependency</td>
<td>41.6*</td>
<td>49.0*</td>
</tr>
<tr>
<td>Word</td>
<td>39.2*</td>
<td>45.2</td>
</tr>
<tr>
<td>Slot</td>
<td>42.1*</td>
<td>49.9*</td>
</tr>
<tr>
<td>Both</td>
<td>43.5* (+15.7%)</td>
<td>53.4* (+17.9%)</td>
</tr>
</tbody>
</table>

All types of relations are useful to infer hidden semantics.

Combining different relations further improves the performance.
CONCLUSIONS

Ontology induction and knowledge graph construction enable systems to automatically acquire open domain knowledge.

MF for SLU provides a principle model that is able to
- unify the automatically acquired knowledge
- adapt to a domain-specific setting
- and then allows systems to consider implicit semantics for better understanding.

The work shows the feasibility and the potential of improving generalization, maintenance, efficiency, and scalability of SDSs.

The proposed unsupervised SLU achieves 43% of MAP on ASR-transcribed conversations.
Q & A

Thanks for your attentions!!