
Yun-Nung (Vivian) Chen, William Yang Wang, and Alexander I. Rudnicky
Overview

- Introduction
- Approach
- Experiments
- Conclusion
In 2011...

- “... I have long been looking for good knowledgeable student to take on the task of getting conversational agents to move beyond prerecorded interaction (or clunky spoken dialog systems) ...”

- “… Wow! Sounds like they know how to achieve the ultimate machine intelligence at CMU...”
Spoken Language Understanding (SLU)

- SLU in dialogue systems
  - SLU maps NL inputs to semantic forms
    - “I would like to go to Shadyside Tuesday.”
      - location: Shadyside  date: Tuesday

- Semantic frames, slots, & values
  - often manually defined by domain experts or developers.

What are the problems?
Problems with Predefined Slots

- **Generalization:** may not generalize to real-world users.
- **Bias propagation:** can bias subsequent data collection and annotation.
- **Maintenance:** when new data comes in, developers need to start a new round of annotation to analyze the data and update the grammar.
- **Efficiency:** time consuming, and high costs.

Can we automatically induce semantic slots only using raw audios?
Overview

Introduction

Approach

Experiments

Conclusion
Probabilistic Frame-Semantic Parsing (Das et al., 2010; 2013) on ASR-transcribeded utterances.

First Step
FrameNet

FrameNet (Baker et al., 1998)

- a linguistically-principled semantic resource, based on the frame-semantics theory.

Example

- **Frame** *(food)*: contains words referring to items of food.
- **Frame Element**: a descriptor indicates the characteristic of food.
  - “low fat milk” →
    - “milk” evokes the “food” frame;
    - “low fat” fills the descriptor FE.
Frame-Semantic Parsing

- **SEMAFOR** (Das et al., 2010; 2013)
  - a state-of-the-art frame-semantics parser, trained on manually annotated FrameNet sentences.
The Panacea?

- Unfortunately...

**Bad!**

*can i have a cheap restaurant*

- Frame: capability
- FT LU: can
- FE LU: i

- Frame: expensiveness
- FT LU: cheap

**Good!**

- Frame: locale by use
- FT/FE LU: restaurant

Task: adapting **generic** frames to **task-specific** settings in dialogue systems.
As A Ranking Problem

- **Main idea**
  - Ranking domain-specific concepts higher than generic semantic concepts

**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

Slot candidate

Slot filler
Slot Ranking Model (1/3)

- Rank the slot candidates by integrating two scores

\[ w(s_i) = \log f(s_i) + \alpha \cdot \log h(s_i) \]

- the frequency of the slot candidate in the SEMAFOR-parsed corpus
- the coherence of slot fillers

- slots with higher frequency may be more important
- domain-specific concepts should focus on fewer topics and be similar to each other

**slot: quantity**

- a
- one
- three
- all

**slot: expensiveness**

- cheap
- inexpensive
- expensive

lower coherence in topic space

higher coherence in topic space
Slot Ranking Model \((2/3)\)

- Measure coherence by **word-level context clustering**
  - For each slot \(s_i\),
    \[
    V(\overline{s_i}) = \{v_1, \ldots, v_j, \ldots, v_J\}
    \]
    slot candidate: expensiveness
corresponding value vectors: “cheap”, “not expensive”
(from the utterances with \(s_i\) in the parsing results)

- We have corresponding cluster vector
  \[
  C(s_i) = \{c_1, \ldots, c_j, \ldots, c_J\} \quad c_j = [c_{j1}, \ldots, c_{jk}, \ldots c_{jK}]\]
  the frequency of words in \(v_j\) clustered into cluster \(k\)

- Measure coherence measure by pair-wised cosine similarity
  \[
  h(s_i) = \frac{\sum_{c_a, c_b \in C(s_i), c_a \neq c_b} \text{CosSim}(c_a, c_b)}{|C(s_i)|^2}
  \]
  The slot with higher \(h(s_i)\) usually focuses on fewer topics, more specific, and preferable for slots of SDS.
Slot Ranking Model (3/3)

Spectral clustering

For each word $w = [r_1, \ldots, r_i, \ldots]$

- $r_i = 1$ when $w$ occurs in the $i$-th utterance
- $r_i = 0$ otherwise

Reasons why spectral clustering:

1) can be solved efficiently by standard linear algebra
2) invariant to the shapes and densities of each cluster
3) projects the manifolds within data into solvable space and often outperform others

The approach can be summarized in five steps:

1. Calculate the distance matrix
2. Derive the affinity matrix
3. Generate the graph Laplacian
4. Eigen decomposition of $L$
5. Perform $K$-means clustering of eigenvectors

Assume that the words in the same utterance are related to each other
Experiments

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 (total # = 10): addr, area, food, name, phone, postcode, price range, signature, task, type
Slot Induction Evaluation

- MAP of the slot ranking model
- measure the quality of induced slots based on induced and reference slots via the mapping table

<table>
<thead>
<tr>
<th>Approach</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASR</td>
</tr>
<tr>
<td>Frequency</td>
<td>67.31</td>
</tr>
<tr>
<td>K-Means</td>
<td>68.45</td>
</tr>
<tr>
<td>Spectral Clustering</td>
<td><strong>69.36</strong></td>
</tr>
</tbody>
</table>

The majority of the reference slots used in a real world SDS can be induced automatically in an unsupervised way.
Slot Filling Evaluation

- MAP of the slot ranking model
  - For each slot, we compute F1 by comparing the lists of extracted and reference slot fillers

The top-5 F1-measure slot-filling corresponding to matched slot mapping for ASR

<table>
<thead>
<tr>
<th>SEMAFORE Slot</th>
<th>Locale by use</th>
<th>Speak on topic</th>
<th>Expensiveness</th>
<th>Origin</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Slot</td>
<td>Type</td>
<td>Addr</td>
<td>Price range</td>
<td>Food</td>
<td>Area</td>
</tr>
<tr>
<td>F1-Hard</td>
<td>89.75</td>
<td>88.86</td>
<td>62.05</td>
<td>36.00</td>
<td>29.81</td>
</tr>
<tr>
<td>F1-Soft</td>
<td>89.96</td>
<td>88.86</td>
<td>62.35</td>
<td>43.48</td>
<td>29.81</td>
</tr>
</tbody>
</table>

F1-Hard: the values of two slot fillers are exactly the same
F1-Soft: the values of two slot fillers both contain at least one overlapping words
Slot Induction and Filling Evaluation

- MAP-F1-Hard / MAP-F1-Soft
  - weight the MAP score with F1-Hard / F1-Soft scores

<table>
<thead>
<tr>
<th>Approach</th>
<th>MAP-F1-Hard</th>
<th>MAP-F1-Soft</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASR</td>
<td>Manual</td>
</tr>
<tr>
<td>Frequency</td>
<td>26.96</td>
<td>27.84</td>
</tr>
<tr>
<td>K-Means</td>
<td>27.38</td>
<td>27.99</td>
</tr>
<tr>
<td>Spectral Clustering</td>
<td><strong>30.52</strong></td>
<td><strong>28.40</strong></td>
</tr>
</tbody>
</table>

When the induced slot mismatches the reference slot, all the slot fillers will be judged as incorrect fillers.
Overview

- Introduction
- Approach
- Experiments
- Conclusion
Conclusion

- We propose an unsupervised approach for automatic induction and filling of semantic slots.

- Our work makes use of a state-of-the-art semantic parser, and adapts the linguistically principled generic FrameNet-style outputs to the target semantic space corresponding to a domain-specific SDS setting.

- Our experiments show that automatically induced semantic slots align well with the reference slots created by domain experts.
Thanks for your attention! 😊