DERIVING LOCAL RELATIONAL SURFACE FORMS FROM DEPENDENCY-BASED ENTITY EMBEDDINGS FOR UNSUPERVISED SPOKEN LANGUAGE UNDERSTANDING

YUN-NUNG (VIVIAN) CHEN & DILEK HAKKANI-TÜR
My Background

Yun-Nung (Vivian) Chen

PhD student advised by Prof. Alexander I Rudnicky

Language Technologies Institute, School of Computer Science, Carnegie Mellon University

Research focus: spoken dialogue system, unsupervised spoken language understanding
Outline

Introduction
  ◦ Main Idea
  ◦ Semantic Knowledge Graph
  ◦ Semantic Interpretation via Relation

Proposed Approach
  ◦ Relation Inference from Gazetteers
  ◦ Relational Surface Form Derivation
  ◦ Probabilistic Enrichment
  ◦ Boostraping

Experiments

Conclusions

Ongoing & Future Work
Main Idea
Relation Detection for Unsupervised SLU

**Spoken Language Understanding (SLU):** convert automatic speech recognition (ASR) outputs into pre-defined semantic output format

"when was james cameron’s avatar released"

Intent: FIND_RELEASE_DATE
Slot-Val: MOVIE_NAME="avatar", DIRECTOR_NAME="james cameron"

**Relation:** semantic interpretation of input utterances
◦ movie.release_date, movie.name, movie.directed_by, director.name

**Unsupervised SLU:** utilize external knowledge to help relation detection **without labelled data**
Semantic Knowledge Graph

Priors for SLU

What are knowledge graphs?
◦ Graphs with
  ◦ strongly typed and uniquely identified entities (nodes)
  ◦ facts/literals connected by relations (edge)

Examples:
◦ Satori, Google KG, Facebook Open Graph, Freebase

How large?
◦ > 500M entities, >1.5B relations, > 5B facts

How broad?
◦ Wikipedia-breadth: “American Football” ⟷ “Zoos”

Semantic Interpretation via Relations

Two Examples
◦ differentiate two examples by including the originating node types in the relation

**User Utterance:**
find movies produced by james cameron

**SPARQL Query (simplified):**

**Logical Form:**
\[ \lambda x. \exists y. \text{movie.produced_by}(x, y) \land \text{person.name}(y, z) \land z="James Cameron" \]

**Relation:**
movie.produced_by \text{ producer.name}

**User Utterance:**
who produced avatar

**SPARQL Query (simplified):**
SELECT ?producer {?movie.name"Avatar". ?movie.produced_by?producer.}

**Logical Form:**
\[ \lambda y. \exists x. \text{movie.produced_by}(x, y) \land \text{movie.name}(x, z) \land z="Avatar" \]

**Relation:**
movie.name \text{ movie.produced_by}
Proposed Framework

“find me some films directed by james cameron”

Background Knowledge
- Entity Dict.
- Relation Inference from Gazetteers
- \( P_E(r \mid w) \)
- Knowledge Graph Entity

Local Relational Surface Form
- Bing Query Snippets
- Entity Embeddings
- Relational Surface Form Derivation
- \( P_F(r \mid w) \)
- Entity Surface Forms
- \( P_C(r \mid w) \)
- Entity Syntactic Contexts

Probabilistic Enrichment
- \( R_u(r) \)

Boostrapping

Relabel

Final Results
Proposed Framework

"find me some films directed by james cameron"

Input Utterance

Background Knowledge

Entity Dict.

Relation Inference from Gazetteers

$P_E(r \mid w)$

Knowledge Graph Entity

Local Relational Surface Form

Bing Query Snippets

Entity Embeddings

Relational Surface Form Derivation

$P_F(r \mid w)$

Entity Surface Forms

$P_C(r \mid w)$

Entity Syntactic Contexts

Probabilistic Enrichment

$R_u(r)$

Boostrapping

Relabel

Final Results
Relation Inference from Gazetteers

Gazetteers (entity lists)

\[
P_E(t_i \mid w) = \frac{C(w, t_i)}{\sum_{t_k \in T(w)} C(w, t_k)}
\]

\[
P_E(r_i \mid w) = P_E(t_i \mid w)
\]

Proposed Framework

“find me some films directed by james cameron”

Input Utterance

Background Knowledge

Entity Dict.

Relation Inference from Gazetteers

$P_E(r | w)$

Knowledge Graph Entity

Probabilistic Enrichment $R_u(r)$

Probabilistic

Boostrapping

Local Relational Surface Form

Bing Query Snippets

Entity Embeddings

Relational Surface Form Derivation

$P_F(r | w)$

Entity Surface Forms

$P_C(r | w)$

Entity Syntactic Contexts

Relabel

Final Results
Avatar is a 2009 American epic science fiction film directed by James Cameron.

Dependency Parsing

Avatar is a 2009 American epic science fiction film directed by James Cameron.
Relational Surface Form Derivation (cont.)
Dependency-Based Entity Embeddings

1) Word & Context Extraction

Word | Contexts
--- | ---
$movie$ | film/nsup
is | film/cop
a | film/det
2009 | film/num
american, epic, science, fiction | film/nn

Word | Contexts
--- | ---
film | film/nsup, is/cop, a/det, 2009/num, american/nn, epic/nn, science/nn, fiction/nn, directed/vmod
directed | $director/prep_by$
$director$ | directed/prep_by
Relational Surface Form Derivation (cont.)

Dependency-Based Entity Embeddings

2) Training Process
   ◦ Each word $w$ is associated with a vector $v_w$ and each context $c$ is represented as a vector $v_c$
   ◦ Learn vector representations for both words and contexts such that the dot product $v_w \cdot v_c$
     associated with good word-context pairs belonging to the training data $D$ is maximized
   ◦ Objective function:
     $$\arg \max_{v_w, v_c} \sum_{(w, c) \in D} \log \frac{1}{1 + \exp(-v_c \cdot v_w)}$$

<table>
<thead>
<tr>
<th>Word</th>
<th>Contexts</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$movie</td>
<td>film/nsub$^{-1}$</td>
</tr>
<tr>
<td>is</td>
<td>film/cop$^{-1}$</td>
</tr>
<tr>
<td>a</td>
<td>film/det$^{-1}$</td>
</tr>
<tr>
<td>2009</td>
<td>film/num$^{-1}$</td>
</tr>
<tr>
<td>american, epic, science, fiction</td>
<td>film/nn$^{-1}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word</th>
<th>Contexts</th>
</tr>
</thead>
<tbody>
<tr>
<td>film</td>
<td>film/nsub, is/cop, a/det, 2009/num, american/nn, epic/nn, science/nn, fiction/nn, directed/vmod</td>
</tr>
<tr>
<td>directed</td>
<td>$$director/prep_by</td>
</tr>
<tr>
<td>$$director</td>
<td>directed/prep_by$^{-1}$</td>
</tr>
</tbody>
</table>
Relational Surface Form Derivation (cont.)

Surface Form Derivation

Entity Surface Forms
- Learn the surface forms corresponding to entities

\[ S_i^F(w_j) = \frac{\text{sim}(w_j, e_i)}{\sum_{e_k \in E} \text{sim}(w_j, e_k)} \]

Entity Syntactic Contexts
- Learn the important contexts of entities

\[ S_i^C(w_j) = \frac{\text{sim}(w_j, e_i)}{\sum_{e_k \in E} \text{sim}(w_j, e_k)} \]

Based on word vector \( v_w \)

Based on context vector \( v_c \)

- \$char\: “character”, “role”, “who”
- \$director\: “director”, “filmmaker”
- \$genre\: “action”, “fiction”

\( \Rightarrow \) with similar contexts

- \$char\: “played”
- \$director\: “directed”

\( \Rightarrow \) frequently occurring together
Proposed Framework

“find me some films directed by james cameron”

Input Utterance

Background Knowledge

Entity Dict. → Relation Inference from Gazetteers → $P_E(r | w)$

Knowledge Graph Entity

Local Relational Surface Form

Bing Query Snippets → Entity Embeddings → Relational Surface Form Derivation → $P_F(r | w)$, $P_C(r | w)$

Entity Surface Forms, Entity Syntactic Contexts

Probabilistic Enrichment $R_u(r)$

Bootstraping

Relabel

Final Results
Probabilistic Enrichment

Integrate relations from
- Prior knowledge $P_E(r \mid w)$
- Entity surface forms $P_F(r \mid w)$
- Entity syntactic contexts $P_C(r \mid w)$

Integrated Relations for Words by
- **Unweighted**: combine all relations with binary values
- **Weighted**: combine all relations and keep the highest weights of relations
- **Highest Weighted**: combine the most possible relation of each word

<table>
<thead>
<tr>
<th>$r$</th>
<th>actor</th>
<th>produced by</th>
<th>location</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_E(r \mid w)$</td>
<td>0.7</td>
<td>0.3</td>
<td>0</td>
</tr>
<tr>
<td>$P_F(r \mid w)$</td>
<td>0.4</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td>$P_C(r \mid w)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

| $R_w(r)$   |         |             |          |
| Unweighted | 1       | 1           | 1        |
| Weighted   | 0.7     | 0.3         | 0.6      |
| Highest Weighted | 0.7 | 0 | 0.6 |

Integrated Relations for Utterances by

$$R_u(r_i) = \max_{w \in u} R_w(r_i)$$

Proposed Framework

“find me some films directed by james cameron”

Input Utterance

Background Knowledge

Entity Dict. → Relation Inference from Gazetteers → \( P_E(r \mid w) \)

Knowledge Graph Entity

Local Relational Surface Form

Bing Query Snippets → Entity Embeddings → Relational Surface Form Derivation → \( P_F(r \mid w) \)

Entity Surface Forms

\( P_C(r \mid w) \)

Entity Syntactic Contexts

Probabilistic Enrichment \( R_u(r) \)

Boostrapping

Relabel

Final Results
Boostrapping
Unsupervised Self-Training

Training a multi-label multi-class classifier estimating relations given an utterance

Utterances with relation weights → creating labels by a threshold

Pseudo labels for training → Adaboost: ensemble M weak classifiers

output prob dist. of relations
Experiments

Dataset

Knowledge Base: Freebase
- 670K entities
- 78 entity types (movie names, actors, etc)

Relation Detection Data
- Crowd-sourced utterances
- Manually annotated with SPARQL queries → relations

<table>
<thead>
<tr>
<th>Query Statistics</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>% entity only</td>
<td>8.9%</td>
<td>10.7%</td>
</tr>
<tr>
<td>% rel only w/ specified movie names</td>
<td>27.1%</td>
<td>27.5%</td>
</tr>
<tr>
<td>% rel only w/ specified other names</td>
<td>39.8%</td>
<td>39.6%</td>
</tr>
<tr>
<td>% more complicated relations</td>
<td>15.4%</td>
<td>14.7%</td>
</tr>
<tr>
<td>% not covered</td>
<td>8.8%</td>
<td>7.6%</td>
</tr>
<tr>
<td>#utterances</td>
<td>3338</td>
<td>1084</td>
</tr>
</tbody>
</table>

User Utterance: who produced avatar
Relation: movie.name movie.produced_by

MOVIE
produced_by
PERSON

name
## Experiments

All performance

Evaluation Metric: micro F-measure (%)

<table>
<thead>
<tr>
<th>Approach</th>
<th>Unweighted</th>
<th>Weighted</th>
<th>Highest Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ori</td>
<td>Bootstrap</td>
<td>Ori</td>
</tr>
<tr>
<td>Gazetteer</td>
<td>35.21</td>
<td>36.91</td>
<td>37.93</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Reg)</td>
<td>34.23</td>
<td>34.91</td>
<td>36.57</td>
</tr>
</tbody>
</table>
Experiments

All performance

<table>
<thead>
<tr>
<th>Approach</th>
<th>Unweighted</th>
<th>Weighted</th>
<th>Highest Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ori</td>
<td>Bootstrap</td>
<td>Ori</td>
</tr>
<tr>
<td>Gazetteer</td>
<td>35.21</td>
<td>36.91</td>
<td>37.93</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Reg)</td>
<td>34.23</td>
<td>34.91</td>
<td>36.57</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Dep)</td>
<td>37.44</td>
<td>38.37</td>
<td>41.01</td>
</tr>
</tbody>
</table>

Baseline

Words derived by dependency embeddings can successfully capture the surface forms of entity tags, while words derived by regular embeddings cannot.
Experiments
All performance

Evaluation Metric: micro F-measure (%)

<table>
<thead>
<tr>
<th>Approach</th>
<th>Unweighted</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ori</td>
<td>Bootstrap</td>
<td>Ori</td>
<td>Bootstrap</td>
<td>Ori</td>
</tr>
<tr>
<td>Gazetteer</td>
<td>35.21</td>
<td>36.91</td>
<td>37.93</td>
<td>40.10</td>
<td>36.08</td>
<td>38.89</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Reg)</td>
<td>34.23</td>
<td>34.91</td>
<td>36.57</td>
<td>38.13</td>
<td>34.69</td>
<td>37.16</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Dep)</td>
<td><strong>37.44</strong></td>
<td><strong>38.37</strong></td>
<td><strong>41.01</strong></td>
<td><strong>41.10</strong></td>
<td>39.19</td>
<td><strong>42.74</strong></td>
</tr>
<tr>
<td>Gazetteer + Entity Context</td>
<td>35.31</td>
<td>37.23</td>
<td>38.04</td>
<td>38.88</td>
<td>37.25</td>
<td>38.04</td>
</tr>
</tbody>
</table>

Baseline

Words derived from entity contexts slightly improve performance.
Experiments
All performance

Evaluation Metric: micro F-measure (%)

<table>
<thead>
<tr>
<th>Approach</th>
<th>Unweighted</th>
<th>Weighted</th>
<th>Highest Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ori</td>
<td>Bootstrap</td>
<td>Ori</td>
</tr>
<tr>
<td>Gazetteer</td>
<td>35.21</td>
<td>36.91</td>
<td>37.93</td>
</tr>
<tr>
<td>Gazetteer + Weakly Supervised</td>
<td>25.07</td>
<td>37.39</td>
<td>39.04</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Reg)</td>
<td>34.23</td>
<td>34.91</td>
<td>36.57</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Dep)</td>
<td>37.44</td>
<td>38.37</td>
<td><strong>41.01</strong></td>
</tr>
<tr>
<td>Gazetteer + Entity Context</td>
<td>35.31</td>
<td>37.23</td>
<td>38.04</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form + Context</td>
<td><strong>37.66</strong></td>
<td><strong>38.64</strong></td>
<td><strong>41.98</strong></td>
</tr>
</tbody>
</table>

Combining all approaches performs best, while the major improvement is from derived entity surface forms.
### Experiments

#### All performance

Evaluation Metric: micro F-measure (%)

<table>
<thead>
<tr>
<th>Approach</th>
<th>Unweighted</th>
<th>Weighted</th>
<th>Highest Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Or</td>
<td>Boostrap</td>
<td>Or</td>
</tr>
<tr>
<td>Gazetteer</td>
<td>35.21</td>
<td>36.91</td>
<td>37.93</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Reg)</td>
<td>34.23</td>
<td>34.91</td>
<td>36.57</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Dep)</td>
<td>37.44</td>
<td>38.37</td>
<td>41.01</td>
</tr>
<tr>
<td>Gazetteer + Entity Context</td>
<td>35.31</td>
<td>37.23</td>
<td>38.04</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form + Context</td>
<td><strong>37.66</strong></td>
<td><strong>38.64</strong></td>
<td><strong>40.29</strong></td>
</tr>
</tbody>
</table>

With the same information, learning surface forms from dependency-based embedding performs better, because there’s mismatch between written and spoken language.
## Experiments

### All performance

**Evaluation Metric: micro F-measure (%)**

<table>
<thead>
<tr>
<th>Approach</th>
<th>Unweighted</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ori</td>
<td>Bootstrap</td>
<td>Ori</td>
<td>Bootstrap</td>
<td>Ori</td>
</tr>
<tr>
<td>Gazetteer</td>
<td>35.21</td>
<td>36.91</td>
<td>37.93</td>
<td>40.10</td>
<td>36.08</td>
<td>38.89</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Reg)</td>
<td>34.23</td>
<td>34.91</td>
<td>36.57</td>
<td>38.13</td>
<td>34.69</td>
<td>37.16</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Dep)</td>
<td>37.44</td>
<td>38.37</td>
<td><strong>41.01</strong></td>
<td>41.10</td>
<td>39.19</td>
<td>42.74</td>
</tr>
<tr>
<td>Gazetteer + Entity Context</td>
<td>35.31</td>
<td>37.23</td>
<td>38.04</td>
<td>38.88</td>
<td>37.25</td>
<td>38.04</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form + Context</td>
<td><strong>37.66</strong></td>
<td><strong>38.64</strong></td>
<td>40.29</td>
<td><strong>41.98</strong></td>
<td><strong>40.07</strong></td>
<td><strong>43.34</strong></td>
</tr>
</tbody>
</table>

Weighted methods perform better when less features, and highest weighted methods perform better when more features.
## Experiments

### All performance

#### Evaluation Metric: micro F-measure (%)

<table>
<thead>
<tr>
<th>Approach</th>
<th>Unweighted</th>
<th>Weighted</th>
<th>Highest Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unweighted</td>
<td>Weighted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ori</td>
<td>Bootstrap</td>
<td>Ori</td>
</tr>
<tr>
<td>Gazetteer</td>
<td>35.21</td>
<td>36.91</td>
<td>37.93</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Reg)</td>
<td>34.23</td>
<td>34.91</td>
<td>36.57</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form (Dep)</td>
<td>37.44</td>
<td>38.37</td>
<td><strong>41.01</strong></td>
</tr>
<tr>
<td>Gazetteer + Entity Context</td>
<td>35.31</td>
<td>37.23</td>
<td>38.04</td>
</tr>
<tr>
<td>Gazetteer + Entity Surface Form + Context</td>
<td><strong>37.66</strong></td>
<td><strong>38.64</strong></td>
<td>40.29</td>
</tr>
<tr>
<td>+ Names of Entity Types</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Additionally adding names of entity types helps improve performance.
Experiments (cont.)
Entity Surface Forms Derived from Dependency Embeddings

The functional similarity carried by dependency-based entity embeddings effectively benefits relation detection task.

<table>
<thead>
<tr>
<th>Entity Tag</th>
<th>Derived Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>$character</td>
<td>character, role, who, girl, she, he, officier</td>
</tr>
<tr>
<td>$director</td>
<td>director, dir, filmmaker</td>
</tr>
<tr>
<td>$genre</td>
<td>comedy, drama, fantasy, cartoon, horror, sci</td>
</tr>
<tr>
<td>$language</td>
<td>language, spanish, english, german</td>
</tr>
<tr>
<td>$producer</td>
<td>producer, filmmaker, screenwriter</td>
</tr>
</tbody>
</table>
Experiments (cont.)

Effectiveness of Boosting

- The best result is the combination of all approaches, because probabilities came from different resources can complement each other.
- Only adding entity surface forms performs similarly, showing that the major improvement comes from relational entity surface forms.
- Boosting significantly improves most performance.
Conclusions

We propose an unsupervised approach to capture the relational surface forms including entity surface forms and entity contexts based on dependency-based entity embeddings.

The detected relations viewed as local observations can be integrated with background knowledge by probabilistic enrichment methods.

Experiments show that involving derived relational surface forms as local cues together with prior knowledge can significantly improve the relation detection task and help open domain SLU.
Ongoing & Future Work
Active Learning

Idea: manually label small data to boost performance

Approach
1. Extract exemplar utterances by clustering
   ◦ Feature set: ngram, relation prob, both
   ◦ Clustering: affinity propagation, k-means, etc.
2. Label exemplar utterances
3. Train the classifier on labelled data

Unsupervised results
   ◦ Embeddings: 0.4334
   ◦ Embeddings + Names: 0.4694

<table>
<thead>
<tr>
<th>#training data (total = 3338)</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline: random selection</td>
<td>0.2892</td>
<td>0.3581</td>
<td>0.3867</td>
<td>0.3921</td>
<td>0.4306</td>
<td>0.4421</td>
<td>0.4522</td>
<td>0.4741</td>
<td>0.4810</td>
<td>0.4821</td>
</tr>
<tr>
<td>Unigram: Euclidean distance</td>
<td>0.1937</td>
<td>0.3167</td>
<td>0.3202</td>
<td>0.3252</td>
<td>0.3557</td>
<td>0.4005</td>
<td>0.4283</td>
<td>0.4447</td>
<td>0.4566</td>
<td>0.4689</td>
</tr>
<tr>
<td>Relation (embeddings)</td>
<td>0.3219</td>
<td>0.3545</td>
<td>0.4126</td>
<td>0.4218</td>
<td>0.4671</td>
<td>0.4907</td>
<td>0.4550</td>
<td>0.4808</td>
<td>0.4629</td>
<td>0.4800</td>
</tr>
<tr>
<td>Relation (names)</td>
<td>0.2780</td>
<td>0.2480</td>
<td>0.3686</td>
<td>0.3966</td>
<td>0.2860</td>
<td>0.4341</td>
<td>0.4490</td>
<td>0.4903</td>
<td>0.5005</td>
<td>0.5150</td>
</tr>
<tr>
<td>Relation (embeddings + names)</td>
<td>0.3457</td>
<td>0.3269</td>
<td>0.4552</td>
<td>0.4012</td>
<td>0.4489</td>
<td>0.4916</td>
<td>0.5191</td>
<td>0.5247</td>
<td>0.5570</td>
<td>0.5417</td>
</tr>
</tbody>
</table>
Thanks to...

Dilek Hakkani-Tür & Gokhan Tur for their advising

Dilek Hakkani-Tür for her help and care

Qi Li & Xiang Li for their discussion

All friends met here for enjoying summer together

MSR for this internship & XBOX ONE
Q & A

THANKS FOR YOUR ATTENTIONS!!