10/02/2014 Sphinx Lunch

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

YUN-NUNG (VIVIAN) CHEN

DILEK HAKKANI-TÜR & GOKHAN TUR
Outline

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

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

Experiments

Conclusions
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” $\leftrightarrow$ “Zoos”

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 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 movie.produced_by
Proposed Framework

“find me some films directed by james cameron”

Input Utterance

Background Knowledge

Entity Dict.
Relation Inference from Gazetteers

Knowledge Graph Entity

Probabilistic Enrichment $R_u(r)$

Boostrapping

Relabel

Final Results

Local Relational Surface Form

Bing Query Snippets
Entity Embeddings
Relational Surface Form Derivation

Entity Surface Forms
Entity Syntactic Contexts

$P_F(r|w)$
$P_C(r|w)$
$P_E(r|w)$
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

$R_u(r)$

Boostraping

Relabel

Final Results

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
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 → \( P_E(r \mid w) \) → Knowledge Graph Entity

Local Relational Surface Form

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

Entity Syntactic Contexts → Probabilistic Enrichment \( R_u(r) \) → Boostraping

Knowledge Graph
Relational Surface Form Derivation
Web Resource Mining

Bing query snippets including entity pairs connected with specific relations in KG

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

Dependency Parsing

directed_by

Avatar

$movie

is

det

a

det

2009

American

epic

science

fiction

film

directed

by

James Cameron

$director
Relational Surface Form Derivation
Dependency-Based Entity Embeddings

1) Word & Context Extraction

<table>
<thead>
<tr>
<th>Word</th>
<th>Contexts</th>
<th>Word</th>
<th>Contexts</th>
</tr>
</thead>
<tbody>
<tr>
<td>$movie</td>
<td>film/nsub⁻¹</td>
<td>film</td>
<td>film/nsub, is/cop, a/det, 2009/num, american/nn, epic/nn, science/nn, fiction/nn, directed/vmod</td>
</tr>
<tr>
<td>is</td>
<td>film/cop⁻¹</td>
<td>2009/num</td>
<td>directed/prep_by</td>
</tr>
<tr>
<td>a</td>
<td>film/det⁻¹</td>
<td>american/nn</td>
<td>$director/prep_by</td>
</tr>
<tr>
<td>2009/num</td>
<td>film/num⁻¹</td>
<td>epic/nn</td>
<td>directed/prep_by⁻¹</td>
</tr>
<tr>
<td>american, epic,</td>
<td>film/nn⁻¹</td>
<td>science/nn</td>
<td></td>
</tr>
<tr>
<td>science, fiction</td>
<td></td>
<td>fiction/nn</td>
<td></td>
</tr>
<tr>
<td>James Cameron</td>
<td></td>
<td>directed</td>
<td></td>
</tr>
<tr>
<td>Avatar</td>
<td></td>
<td>directed</td>
<td></td>
</tr>
<tr>
<td>is</td>
<td></td>
<td>directed</td>
<td></td>
</tr>
<tr>
<td>is</td>
<td></td>
<td>directed</td>
<td></td>
</tr>
<tr>
<td>is</td>
<td></td>
<td>directed</td>
<td></td>
</tr>
<tr>
<td>is</td>
<td></td>
<td>directed</td>
<td></td>
</tr>
<tr>
<td>is</td>
<td></td>
<td>directed</td>
<td></td>
</tr>
<tr>
<td>is</td>
<td></td>
<td>directed</td>
<td></td>
</tr>
</tbody>
</table>
Relational Surface Form Derivation
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>
<th>Word</th>
<th>Contexts</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$movie</td>
<td>film/nsub⁻¹</td>
<td>film</td>
<td>film/nsub, is/cop, a/det, 2009/num, american/nn, epic/nn, science/nn, fiction/nn, directed/vmod</td>
</tr>
<tr>
<td>is</td>
<td>film/cop⁻¹</td>
<td></td>
<td>directed</td>
</tr>
<tr>
<td>a</td>
<td>film/det⁻¹</td>
<td></td>
<td>$$director/prep_by</td>
</tr>
<tr>
<td>2009</td>
<td>film/num⁻¹</td>
<td></td>
<td>$$director/prep_by⁻¹</td>
</tr>
<tr>
<td>american, epic,</td>
<td>film/nn⁻¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>science, fiction</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Relational Surface Form Derivation

Surface Form Derivation

Entity Surface Forms

- learn the surface forms corresponding to entities

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

\[ P^F (r_{i|w_j}) \]

Entity Syntactic Contexts

- learn the important contexts of entities

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

\[ P^C (r_{i|w_j}) \]

$char, $director, etc.

$char$: “character”, “role”, “who”

$director$: “director”, “filmmaker”

$genre$: “action”, “fiction”

$\rightarrow$ with similar contexts

$char$: “played”

$director$: “directed”

$\rightarrow$ frequently occurring together

based on word vector $v_w$

based on context vector $v_c$
Proposed Framework

“find me some films directed by james cameron”

Input Utterance

Knowledge Graph

Entity Dict.

Relation Inference from Gazetteers

Knowledge Graph Entity

Probabilistic Enrichment

R_u(r)

Boosting

Relabel

Final Results

Background Knowledge

Local Relational Surface Form

Entity Surface Forms

Entity Syntactic Contexts

Bing Query Snippets

Entity Embeddings

Relational Surface Form Derivation

P_E(r | w)

P_F(r | w)

P_C(r | w)
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

### Integrated Relations for Utterances by

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

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

Unweighted $R_w(r)$
Weighted $R_w(r)$
Highest Weighted $R_w(r)$

Proposed Framework

“find me some films directed by james cameron”

Input Utterance

Background Knowledge

Entity Dict.
Relation Inference from Gazetteers

Knowledge Graph Entity

Probabilistic Enrichment $R_u(r)$

Boostrapping

Final Results

Local Relational Surface Form

Bing Query Snippets

Entity Embeddings

Entity Surface Form Derivation

Entity Surface Forms

Entity Syntactic Contexts

Relabel

Knowledge Graph
Boostrapping
Unsupervised Self-Training

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

Utterances with relation weights:
- $R_{u1}(r)$
- $R_{u2}(r)$
- $R_{u3}(r)$

Creating labels by a threshold.

Pseudo labels for training:
- $u_1: L_{u1}(r)$
- $u_2: L_{u2}(r)$
- $u_3: L_{u3}(r)$

Adaboost: ensemble $M$ weak classifiers.

Classifier output: prob dist. of relations.
Experiments of Relation Detection

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

PERSON

produced_by

name
Experiments of Relation Detection

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>

Baseline
Experiments of Relation Detection
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>
<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 of Relation Detection
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>
<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>
</tbody>
</table>

Baseline

Words derived from entity contexts slightly improve performance.
Experiments of Relation Detection
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>41.01</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>37.66</td>
<td>38.64</td>
<td>40.29</td>
<td>41.98</td>
<td>40.07</td>
<td>43.34</td>
</tr>
</tbody>
</table>

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

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>Ori</td>
<td>Bootstrap</td>
<td>Ori</td>
<td>Bootstrap</td>
<td>Ori</td>
<td>Bootstrap</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>41.01</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>37.66</td>
<td>38.64</td>
<td>40.29</td>
<td>41.98</td>
<td>40.07</td>
<td>43.34</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 of Relation Detection

### All performance

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

<table>
<thead>
<tr>
<th>Approach</th>
<th>Unweighted</th>
<th></th>
<th>Weighted</th>
<th></th>
<th>Highest Weighted</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>41.01</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>37.66</td>
<td>38.64</td>
<td>40.29</td>
<td>41.98</td>
<td>40.07</td>
<td>43.34</td>
</tr>
</tbody>
</table>

Weighted methods perform better when less features, and highest weighted methods perform better when more features.
Experiments of Relation Detection
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 of Relation Detection
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.
Q & A 😊

THANKS FOR YOUR ATTENTIONS!!