Connecting the Dots: Event Graph Schema Induction with Path Language Modeling

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, Clare Voss
Event Schema Induction Related Work

• How to capture complex connections among events?
  • Temporal relations exist between almost all events, even those that are not semantically related
  • Causal relations have been hobbled by low inter-annotator agreement (Hong et al., 2016).
• Two events are connected through entities and their relations

Previous Work:
Event Narrative Chain (Chambers and Jurafsky, 2008, 2009, 2010)

Our Paper:
Event Narrative Graph Schema
Problem Formulation

- **Input**: Instance graphs, where each node is an entity or event, each edge is an argument role or relation.
- **Output**: Graph schemas between two event types, where each node is an entity type or event type.
- Instance graphs (a) and (b) refer to very different event instances, but they both illustrate a same scenario.

---

**Event Instance Graphs**

(a)

- Transport
- Deploy
- Destination
- Fac
- Sevastopol
- GPE
- Ukraine
- PER
- Troops
- Artifact
- Affiliation
- Agent
- Origin
- WEa
- Tank
- Instrument
- Attack
- Attack

(b)

- Transport
- Carry
- Destination
- Fac
- Maidan Square
- GPE
- Kyiv
- PER
- Police
- GPE
- Ukraine
- WEa
- Stone
- Place
- Affiliation
- Target
- Part-Whole
- Located_In
- Attack
- Hit

**Graph Schema Induction**

- Transport
- LOC
- WEa
- Instrument
- Attack
- Fac
- Destination
- Origin
- Part-Whole
- GPE
- Place
- Affiliation
- Target
- PER
- Police
- GPE
- Ukraine
- WEa
- Stone
- Place
- Affiliation
- Target
Event Graph Schema Induction Framework

- We select **salient** and **coherent** paths based on Path Language Model, and merge them into graph schemas.

---

**Event Instance Graphs**

- **(a)**
  - Transport: deploy, destination
  - FAC: Sevastopol
  - GPE: Russia
  - Affiliations and roles:
    - **artifacts**: artifact
    - **agents**: agent
    - **WEA**: tank
  - Path: attacker

- **(b)**
  - Transport: carry, destination
  - FAC: Maidan Square
  - GPE: Kyiv
  - Affiliations and roles:
    - **artifacts**: artifact, hit
    - **agents**: protesters
    - **WEA**: stone
  - Path: attacker

---

**Path Language Model**

- Transport: origin, destination
- FAC: destination
- GPE: located_in
- WEA: place
- Attack: target

---

**Graph Schema Induction**

- Affiliations and roles:
  - **artifacts**: artifact
  - **agents**: attacker
  - **WEA**: instrument
  - **LOC**: place
  - **GPE**: part-whole
  - **PER**: part-whole
  - **ORG**: target
A good graph schema for two event types consists of **salient** and **coherent** paths between them.

- **Salience**: recurring event-event connection patterns
- **Coherence**: semantically coherent

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Examples</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Path</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salience</td>
<td>High</td>
<td><img src="image1" alt="Graph" /></td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td><img src="image2" alt="Graph" /></td>
</tr>
<tr>
<td>Semantic Coherence</td>
<td>High</td>
<td><img src="image3" alt="Graph" /></td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td><img src="image4" alt="Graph" /></td>
</tr>
<tr>
<td>Multiple Paths</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic Consistency</td>
<td>High</td>
<td><img src="image5" alt="Graph" /></td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td><img src="image6" alt="Graph" /></td>
</tr>
</tbody>
</table>
Path Language Model

- Path Language Model is trained on two tasks
  - Autoregressive Language Model Loss: capturing the frequency and coherence of a single path
  - Neighbor Path Classification Loss: capturing co-occurrence of two paths
Experiment: Instance Coverage

- A salient schema can serve as a skeleton to recover instance graphs
  - We use each graph schema to match back to each ground-truth instance graph and evaluate their intersection in terms of Precision and Recall.

\[
\text{Precision} = \frac{\sum_{s \in S} \sum_{g \in G} |g \cap s|^s}{\sum_{s \in S} |s|^s}
\]

\[
\text{Recall} = \frac{\sum_{s \in S} \sum_{g \in G} |g \cap s|^1}{\sum_{g \in G} |g|^1}
\]
Experiment: Instance Coverage

- We evaluate the intersection by substructures of the schema graph, i.e., paths of different lengths.

- Proves ability of PathLM:
  - Higher gains on longer path queries (e.g. \( l = 7 \)) → able to capture complex graph structures involving long distance between related events.
  - Larger gains compared to baselines on Schema@10 than Schema@20, demonstrating the effectiveness of our ranking approach, especially on top ranked ones.
  - Schemas induced from automatically constructed and manually constructed event graph instances have comparable performance → robust to extraction noise.

![Instance Coverage on Paths of \( l = 5 \)](image1)

![Instance Coverage on Paths of \( l = 7 \)](image2)
Experiment: Instance Coverage

• We evaluate the intersection by substructures of the schema graph, i.e., paths of different lengths
• Proves ability of PathLM
  • Higher gains on longer path queries (e.g. \( l = 7 \)) \( \rightarrow \) able to capture complex graph structures involving long distance between related events.
  • Larger gains compared to baselines on Schema@10 than Schema@20, demonstrating the effectiveness of our ranking approach, especially on top ranked ones.
  • Schemas induced from automatically constructed and manually constructed event graph instances have comparable performance \( \rightarrow \) robust to extraction noise
Experiment: Instance Coverage

• We evaluate the intersection by substructures of the schema graph, i.e., paths of different lengths

• Proves ability of PathLM:
  • Higher gains on longer path queries (e.g. \( l = 7 \)) \( \rightarrow \) able to capture complex graph structures involving long distance between related events.
  • Larger gains compared to baselines on Schema@10 than Schema@20, demonstrating the effectiveness of our ranking approach, especially on top ranked ones.
  • Schemas induced from automatically constructed and manually constructed event graph instances have comparable performance \( \rightarrow \) robust to extraction noise

---

### INSTANCE COVERAGE ON PATHS OF \( L = 5 \)

<table>
<thead>
<tr>
<th>F(_1) (%)</th>
<th>Frequency</th>
<th>TrigramLM</th>
<th>PathLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema@10 (Annotation)</td>
<td>85</td>
<td>70</td>
<td>90</td>
</tr>
<tr>
<td>Schema@10 (System)</td>
<td>60</td>
<td>50</td>
<td>70</td>
</tr>
<tr>
<td>Schema@20 (Annotation)</td>
<td>80</td>
<td>65</td>
<td>85</td>
</tr>
<tr>
<td>Schema@20 (System)</td>
<td>50</td>
<td>40</td>
<td>60</td>
</tr>
</tbody>
</table>

### INSTANCE COVERAGE ON PATHS OF \( L = 7 \)

<table>
<thead>
<tr>
<th>F(_1) (%)</th>
<th>Frequency</th>
<th>TrigramLM</th>
<th>PathLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema@10 (Annotation)</td>
<td>70</td>
<td>60</td>
<td>80</td>
</tr>
<tr>
<td>Schema@10 (System)</td>
<td>50</td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>Schema@20 (Annotation)</td>
<td>75</td>
<td>65</td>
<td>85</td>
</tr>
<tr>
<td>Schema@20 (System)</td>
<td>55</td>
<td>45</td>
<td>70</td>
</tr>
</tbody>
</table>
 Experiment: Instance Coherence

- we hypothesize that an instance graph between two events 𝑣 and 𝑣’ is coherent if 𝑣 and 𝑣’ are from the same discourse (e.g., a news document)
- We carefully select 24 documents with each document talking about a unique complex event such as *Iraq War* or *North Korea Nuclear Test*
- We define **Instance Coherence** as the proportion of in-doc path instances

\[
\text{Coherence} = \frac{\sum_{s \in S} \sum_{g \in G} \sum_{p \in g \cap s} f(p) \cdot \mathbb{I}_g}{\sum_{s \in S} \sum_{g \in G} \sum_{p \in g \cap s} f(p)}.
\]

<table>
<thead>
<tr>
<th>Historical Models</th>
<th>Schema@10 (%)</th>
<th>Schema@20 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>67.8</td>
<td>65.6</td>
</tr>
<tr>
<td>UnigramLM</td>
<td>62.4</td>
<td>69.9</td>
</tr>
<tr>
<td>BigramLM</td>
<td>59.0</td>
<td>67.5</td>
</tr>
<tr>
<td>TrigramLM</td>
<td>56.6</td>
<td>64.9</td>
</tr>
<tr>
<td>PathLM</td>
<td>76.0</td>
<td>79.9</td>
</tr>
<tr>
<td>w/o CLS_NP</td>
<td>75.3</td>
<td>79.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>60.1</td>
<td>65.6</td>
</tr>
<tr>
<td>UnigramLM</td>
<td>61.8</td>
<td>70.0</td>
</tr>
<tr>
<td>BigramLM</td>
<td>59.7</td>
<td>69.6</td>
</tr>
<tr>
<td>TrigramLM</td>
<td>55.8</td>
<td>65.8</td>
</tr>
<tr>
<td>PathLM</td>
<td>76.4</td>
<td>78.5</td>
</tr>
<tr>
<td>w/o CLS_NP</td>
<td>73.9</td>
<td>77.1</td>
</tr>
</tbody>
</table>
Schema-Guided Information Extraction

- Use the state-of-the-art IE system OneIE (Lin et al., 2020) to decode converts each input document into an IE graph.
- Each path in the graph schema is encoded as a single global feature for scoring candidate IE graphs.
- OneIE promotes candidate IE graphs containing paths matching schema graphs.
- http://blender.cs.illinois.edu/software/oneie
- F-scores (%) on ACE2005 data [Lin et al., ACL2020]:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Entity</th>
<th>Event Trigger Identification</th>
<th>Event Trigger Classification</th>
<th>Event Argument Identification</th>
<th>Event Argument Classification</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>90.3</td>
<td>75.8</td>
<td>72.7</td>
<td>57.8</td>
<td>55.5</td>
<td>44.7</td>
</tr>
<tr>
<td>+PathLM</td>
<td>90.2</td>
<td>76.0</td>
<td>73.4</td>
<td>59.0</td>
<td>56.6</td>
<td>60.9</td>
</tr>
</tbody>
</table>
THANK YOU!