Brief Introduction and Review of Open Information Extraction (Open-IE) Systems

CS 290D Project Presentation

Presenter: Sina Miran

* All of the materials presented are research outcomes of Turing Center at UW supervised by Prof. Oren Etzioni
Outline

1. Introduction to IE, Open-IE, and Machine Reading
2. TextRunner
3. ReVerb
4. OLLIE
5. Comparison and Brief Performance Analysis
6. References
Information Extraction Systems: Distill semantic relations from natural language text.

Machine Reading: (Open) Information Extraction + Inference

Sample Output of IE: \( \text{IE(sentence)} = \text{Relation instance, probability} \)

“Edison was the inventor of the light bulb.”
Output: (Edison, invented, light bulb), 0.9

Traditional IE Systems:
1. Hand labeled data (documents on a special topic)
2. Looking for pre-specified relations (supervised learning)

Limited because of their supervised nature and can’t scale to world wide web
### Open-IE vs. Traditional IE

<table>
<thead>
<tr>
<th>Input</th>
<th>Traditional IE</th>
<th>Open IE</th>
</tr>
</thead>
<tbody>
<tr>
<td>corpus + hand-labeled data</td>
<td>corpus</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relations</th>
<th>Traditional IE</th>
<th>Open IE</th>
</tr>
</thead>
<tbody>
<tr>
<td>specified in advance</td>
<td>discovered</td>
<td>automatically</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Extractor</th>
<th>Traditional IE</th>
<th>Open IE</th>
</tr>
</thead>
<tbody>
<tr>
<td>relation specific</td>
<td>relation</td>
<td>independent</td>
</tr>
</tbody>
</table>

Open-IE: Avoid hand-labeling sentences (documents), Single pass over corpus, No pre-specified vocabulary
2. TextRunner

TextRunner Architecture:

Identifying relations using linear Conditional Random Fields (CRF)

Tuple \( t = (e_i, r_{ij}, e_j) \)

Tuple correction and inference (explained next)
Inference and tuple correction:

(X, born in, 1941)         (Y, born in, 1941)         P ( X = Y ) determined by shared relations
(X, citizen of, US)          (Y, citizen of, US)
(X, friend of, Joe)      (Y, friend of, Joe)

(1, R1, 2)         (1, R2, 2)
(2, R1, 4)         (2, R2, 4)         P ( R1 = R2 ) determined by shared argument pairs
(4, R1, 8)         (4, R2, 8)

Next: How likely is an extraction to be correct!?
Formal Problem Statement (for inference and correction):

If an extraction $x$ appears $k$ times in a set of $n$ distinct sentences each suggesting that $x$ belongs to $C$, what is the probability that $x \in C$?

$C$ is a class such as “cities” or a relation such as “mayor of”

We only count distinct sentences!

$$P(x \in C | x \text{ appears } k \text{ times in } n \text{ draws}) = \frac{\sum_{r \in \text{num}(C)} \left( \frac{r}{\delta} \right)^k \left(1 - \frac{r}{\delta} \right)^{n-k}}{\sum_{r' \in \text{num}(C \cup E)} \left( \frac{r'}{\delta} \right)^k \left(1 - \frac{r'}{\delta} \right)^{n-k}}$$

Increase exponentially with $k$ and decrease exponentially with $n$. 

Open Information Extraction (Open-IE)
3. ReVerb

Problems with TextRunner that motivated ReVerb:

<table>
<thead>
<tr>
<th>is</th>
<th>is an album by, is the author of, is a city in</th>
</tr>
</thead>
<tbody>
<tr>
<td>has</td>
<td>has a population of, has a Ph.D. in, has a cameo in</td>
</tr>
<tr>
<td>made</td>
<td>made a deal with, made a promise to</td>
</tr>
<tr>
<td>took</td>
<td>took place in, took control over, took advantage of</td>
</tr>
<tr>
<td>gave</td>
<td>gave birth to, gave a talk at, gave new meaning to</td>
</tr>
<tr>
<td>got</td>
<td>got tickets to, got a deal on, got funding from</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Incoherent Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>The guide <em>contains</em> dead links and <em>omits</em> sites.</td>
<td>contains omits</td>
</tr>
<tr>
<td>The Mark 14 <em>was central</em> to the <em>torpedo</em> scandal of the fleet.</td>
<td>was central torpedo</td>
</tr>
<tr>
<td>They <em>recalled</em> that Nungesser <em>began</em> his career as a precinct leader.</td>
<td>recalled began</td>
</tr>
</tbody>
</table>
3. ReVerb

Indentify Relations from Verbs.

1) Find longest phrase matching a simple syntactic constraint:

\[
V \mid VP \mid VWP
\]

\[
V = \text{verb}
\]

\[
W = (\text{noun} \mid \text{adj} \mid \text{adv} \mid \text{pron} \mid \text{det})
\]

\[
P = (\text{prep} \mid \text{particle} \mid \text{inf. marker})
\]

2) Constraint: \(| \text{args(Relation)} | > k\)
Which means we have to see a valid Relation at least \(k\) times (\(k \sim 20\))

Why learn the relations and use machine learning when we know most of them take specific forms?!
4. OLLIE

Problems with ReVerb that motivated OLLIE (Open Language Learning for IE)

1) Limited to relations that are mediated by verbs

2) Ignores context which results in extracting tuples that are not factual

OLLIE covers a larger number of relation expressions, and expands Open-IE representation to allow additional context information such as attribution and clause modifiers

1. “After winning the Superbowl, the Saints are now the top dogs of the NFL.”
   O: (the Saints; win; the Superbowl)
2. “There are plenty of taxis available at Bali airport.”
   O: (taxi; be available at; Bali airport)
3. “Microsoft co-founder Bill Gates spoke at...”
   O: (Bill Gates; be co-founder of; Microsoft)
4. “Early astronomers believed that the earth is the center of the universe.”
   R: (the earth; be the center of; the universe)
   W: (the earth; be; the center of the universe)
   O: ((the earth; be the center of; the universe)
      AttributedTo believe; Early astronomers)
5. “If he wins five key states, Romney will be elected President.”
   R,W: (Romney; will be elected; President)
   O: ((Romney; will be elected; President)
      ClausalModifier if; he wins five key states)
4. OLLIE

We assume each relation in the corpus has an equivalent ReVerb tuple!

Learning Open Patterns:

1) Extract the high confidence tuples from ReVerb.
2) For each tuple, find all sentences in the corpus containing the words in the tuple.
3) Using a dependency parser specify the patterns corresponding to each ReVerb tuple selected.
4. OLLIE

Pattern Matching:

Using the output of a dependency parser match the learned open patterns to the sentences

Context Analysis:

Using dependency parser of a sentence to add additional fields like AttributedTo or ClauseModifier
5. Comparison and Brief Performance Analysis

ReVerb vs. TextRunner

Dataset: 500 sentences sampled from the web using Yahoo’s random link services

Two human judges evaluated the extractions as correct or false
5. Comparison and Brief Performance Analysis

OLLIE and ReVerb

- Dataset: 300 random sentences from three random sources.
- Considered approximately 2000 extractions from each system
- Extractions were annotated manually as correct or false

- Dataset: 100M random sentences from the ClueWeb corpus
6. References


Questions?!