CS544: Paraphrase Acquisition

March 1, 2012

Zornitsa Kozareva
USC/ISI
Marina del Rey, CA
kozareva@isi.edu
www.isi.edu/~kozareva

Machine Translation

(Mirkin et al., 2009)

He abhorred the men’s unctuous ways.
He disliked the men’s flattering ways.

Source | Target
--------|--------
hated the man | hasste die Männer
flattering ways | schmeichelhafte Weise
...... | ......

Er liebte die Männer unctuous Weise
Er hasste die Männer schmeichelhafte Weise
Dependency Tree

• A sentences can be represented by a set of dependency relations that form a tree

John found a solution to the problem.

run Minipar
dependency parser

John

subj

found

obj

a

det

solution

mod
to

pcomp

problem

the

det

• A word in the sentence can have several modifiers
• A head word is the governor, a modifier is the dependent
• The root of the dependency tree does not modify any word
Dependency Tree

John found a solution to the problem.

• Links represent dependency relations
• The direction of the links is from head to modifier in the relation

Example of Minipar’s Dependency Relations

<table>
<thead>
<tr>
<th>Relation</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>subj</td>
<td>subject of a verb</td>
<td>John loves Mary.</td>
</tr>
<tr>
<td>det</td>
<td>determiner of a noun</td>
<td>the dog</td>
</tr>
<tr>
<td>gen</td>
<td>genitive modifier of a noun</td>
<td>John’s dog</td>
</tr>
<tr>
<td>mod</td>
<td>adjunct modifier of any head</td>
<td>tiny hole</td>
</tr>
<tr>
<td>nn</td>
<td>prenominal modifier of noun</td>
<td>station manager</td>
</tr>
<tr>
<td>appo</td>
<td>appositive of a noun</td>
<td>the CEO, John</td>
</tr>
</tbody>
</table>
Paths in Dependency Tree

- Representation

\[ \text{John} \quad \text{found} \quad \text{a} \quad \text{solution} \quad \text{to} \quad \text{the} \quad \text{problem} \]

- In Minipar, each link between two words in the dependency tree represents a direct semantic relationship.

- However, a path allows to capture indirect semantic relationships between two content words.

Paths in Dependency Tree

- A path concatenates the dependency relations and words along the path.

Ex: What is the path between \textit{John} and \textit{problem}?


Paths in Dependency Tree

John found a solution to the problem.

- A path concatenates the dependency relations and words along the path

\[ N:subj<V\text{find}-V:obj:N\rightarrow solution\rightarrow N:to:N \]

Filler for SlotX is \textit{John}

Filler for SlotY is \textit{problem}

Paths in Dependency Tree

John found a solution to the problem.

- A path concatenates the dependency relations and words along the path

\[ N:subj<V\text{find}-V:obj:N\rightarrow solution\rightarrow N:to:N \]

- Dependency relations that are not slots are called internal relations

Ex. \[ \text{find}\rightarrow V:obj:N\rightarrow solution \]
Discovering Inference Rules From Text

- Extract all paths in a corpus together with the slot fillers
- Measure the relatedness of a slot filler and a path
- Measure the similarity between two paths
  (the more features two paths share, the more similar they are)
- Use on the Extended Distributional Hypothesis

If two paths tend to occur in similar contexts,
the meaning of the paths tends to be similar

Constraints for Path Extraction

- **Slot fillers** must be nouns or pronouns, because in the paraphrasing task the slots correspond to variables which are instantiated by entities
- Any dependency relation which does not connect two content words (noun, verb, adjective or adverb) is excluded from the path (won’t extract a relation between John and a)
- An internal relation must be between a verb and an object-noun (because in paraphrasing the center must be a verb)
Example of Extracted Paths

Sam had previously bought bighorn sheep from Comstock.

<table>
<thead>
<tr>
<th>Extracted Path</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>N:subj:V&lt;-buy-&gt;V:from:N</td>
<td>X buy something from Y</td>
</tr>
<tr>
<td>N:subj:V&lt;-buy-&gt;V:obj:N</td>
<td>X buys Y</td>
</tr>
<tr>
<td>N:nn:N&lt;-sheep&lt;-N:obj:V&lt;-buy-&gt;V:from:N</td>
<td>X sheep is bought from Y</td>
</tr>
<tr>
<td>N:obj:V&lt;-buy-&gt;V:from:N</td>
<td>X is bought from Y</td>
</tr>
</tbody>
</table>

Example of Slot Fillers for Two Paths

<table>
<thead>
<tr>
<th>“X finds a solution to Y”</th>
<th>“X solves Y”</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SlotX</strong></td>
<td><strong>SlotY</strong></td>
</tr>
<tr>
<td>commission</td>
<td>strike</td>
</tr>
<tr>
<td>committee</td>
<td>civil war</td>
</tr>
<tr>
<td>government</td>
<td>crisis</td>
</tr>
<tr>
<td>government</td>
<td>problem</td>
</tr>
<tr>
<td>he</td>
<td>situation</td>
</tr>
<tr>
<td>i</td>
<td>budget deficit</td>
</tr>
<tr>
<td>legislator</td>
<td>dispute</td>
</tr>
<tr>
<td>sheriff</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Discovering Inference Rules From Text

- Extract all paths in a corpus together with the slot fillers
- Measure the relatedness of a slot filler and a path

Mutual Information

- Mutual Information (MI) is between two events
  \[ \text{PMI}(w_1, w_2) = \log_2 \frac{p(w_1, w_2)}{p(w_1) \times p(w_2)} \]
- But, in the paraphrase model we have three events: \textit{path, slot} and \textit{slot filler}
  \[ \text{PMI}(\text{pattern, slot, filler}) = \log_2 \frac{P(\text{pattern, slot, filler})}{P(\text{slot}) \times P(\text{pattern | slot}) \times P(\text{filler | slot})} \]

This model assumes that the path and the filler are conditionally independent given a slot.
MI for a Path-Slot-Filler Triple

\[
PMI(pattern, slot, filler) = \log \frac{P(pattern, slot, filler)}{P(slot) \cdot P(pattern | slot) \cdot P(filler | slot)}
\]

\[
PMI(pattern, slot, filler) = \log \frac{|pattern, slot, filler|}{|*,*,*| \cdot |pattern, slot,*| \cdot |*,slot,*|}
\]

\[
= \log \frac{|pattern,slot,filler| \times |*,slot,*|}{|pattern,slot,*| \times |*,*,*|}
\]

**Generated Triples**

<table>
<thead>
<tr>
<th>SlotX</th>
<th>Freq</th>
<th>MI</th>
<th>SlotY</th>
<th>Freq</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td>driver</td>
<td>1</td>
<td>2.45</td>
<td>bus</td>
<td>2</td>
<td>3.09</td>
</tr>
<tr>
<td>equipment</td>
<td>1</td>
<td>1.65</td>
<td>coach</td>
<td>1</td>
<td>2.05</td>
</tr>
<tr>
<td>police</td>
<td>2</td>
<td>2.24</td>
<td>debris</td>
<td>1</td>
<td>2.36</td>
</tr>
<tr>
<td>rescuer</td>
<td>3</td>
<td>4.84</td>
<td>feet</td>
<td>1</td>
<td>1.75</td>
</tr>
<tr>
<td>resident</td>
<td>1</td>
<td>1.60</td>
<td>hut</td>
<td>1</td>
<td>2.73</td>
</tr>
<tr>
<td>who</td>
<td>2</td>
<td>1.32</td>
<td>landslide</td>
<td>1</td>
<td>2.39</td>
</tr>
<tr>
<td>worker</td>
<td>1</td>
<td>1.37</td>
<td>metal</td>
<td>1</td>
<td>2.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>wreckage</td>
<td>3</td>
<td>4.81</td>
</tr>
</tbody>
</table>
Discovering Inference Rules From Text

✓ Extract all paths in a corpus together with the slot fillers
✓ Measure the relatedness of a slot filler and a path

• Measure the similarity between two paths
  (the more features two paths share, the more similar they are)

\[
\text{sim}(slot_1, slot_2) = \frac{\sum_{w \in T(p_1, s) \cap T(p_2, s)} \text{mi}(p_1, s, w) + \text{mi}(p_2, s, w)}{\sum_{w \in T(p_1, s)} \text{mi}(p_1, s, w) + \sum_{w \in T(p_2, s)} \text{mi}(p_2, s, w)}
\]

\(p1\) and \(p2\) are paths
\(s\) is a slot
\(T(pi, s)\) is a set of words that fill in the \(s\) slot for path \(pi\)
Generated Triples

| "X pulls body from Y" | MI | "X retrieves Y" | MI |
|----------------------|----|----------------|--|--|
| SlotX                | MI | SlotY          | MI |
| driver               | 2.45 | bus           | 3.09 |
| equipment            | 1.65 | coach         | 2.05 |
| police               | 2.24 | debris        | 2.36 |
| rescuer              | 4.84 | feet          | 1.75 |
| resident             | 1.60 | hut           | 2.73 |
| who                  | 1.32 | landslide     | 2.39 |
| worker               | 1.37 | metal         | 2.09 |
|                     |  | wreckage      | 4.81 |

Similarity Between Paths

\[ sim(p_1, p_2) = \sqrt{\frac{sim(SlotX_1, SlotX_2) \times sim(SlotY_1, SlotY_2)}} \]

\( p_1 \) and \( p_2 \) are paths defined by the geometric average of similarities of their SlotX and SlotY.
Finding the Most Similar Path

- Given a path, find the most similar path
  - For every word, store its SlotX and SlotY in all paths
  - Form candidate paths
  - Count the number of features between these paths

---

Question answering

What does the Peugeot company manufacture?

The Peugeot company manufactures a product.

RETE SYSTEM

REFORMULATOR

RETRIEVAL COMPONENT

Document Collection

1: Answer
2: Answer
3: Answer
4: Answer
5: Answer

Hickl\&Harabagiu
### Question: What does the Peugeot company manufacture?

**Path:** X manufactures Y

### Manual Paraphrases:
- X makes Y
- X produce Y
- X is in Y business
- Y is manufactured by X
- Y is provided by X
- Y is X's product
- Y is product from X
- Y is X product
- Y is product made by X
- Y is example of X product
- X is manufacturer of Y
- Find Y in X's product line
- Find Y in X catalog

### Automated Paraphrases:
- X produces Y
- X markets Y
- X develops Y
- X is supplier of Y
- X ships Y
- X supplies Y
- Y is manufactured by X
- X is a maker of Y
- X introduces Y
- X exports Y
- X makes Y
- X builds Y
- X's production of Y
- Y is bought from X
- X's line of Y
- X assembles Y
- X is Y maker
- X's Y factory
- X is manufacturer of Y

---

**Quick Task:**

1) Write your own paraphrase rules.
2) Who generated more rules the human or the machine?
3) How well do you think the paraphrases generated by the machine are?
**Evaluation**

<table>
<thead>
<tr>
<th>Question</th>
<th>Path</th>
<th>Manual</th>
<th>Automated</th>
<th>Intersection</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>What does the Peugeot company manufacture?</td>
<td>X manufacture Y</td>
<td>13</td>
<td>37</td>
<td>4</td>
<td>92.5%</td>
</tr>
<tr>
<td>Who is the author of the book, “The Iron Lady: A Biography of Margaret Thatcher”?</td>
<td>X is author of Y</td>
<td>7</td>
<td>21</td>
<td>2</td>
<td>52.5%</td>
</tr>
<tr>
<td>Why did David Koresh ask the FBI for a word processor?</td>
<td>X asks Y</td>
<td>2</td>
<td>23</td>
<td>0</td>
<td>57.5%</td>
</tr>
</tbody>
</table>

**More Examples**

<table>
<thead>
<tr>
<th>Question</th>
<th>Path</th>
<th>Manual Paraphrases</th>
<th>Automated Paraphrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who is the author of the book, “The Iron Lady: A Biography of Margaret Thatcher”?</td>
<td>X is author of Y</td>
<td>Y is the work of X X is the writer of Y X penned Y X produced Y X authored Y X chronicled Y X wrote Y</td>
<td>X co-authors Y X is co-author of Y X writes Y X edits Y Y is co-authored by X Y is authored by X X tells story in Y X translates Y X writes in Y X notes in Y</td>
</tr>
</tbody>
</table>
Next Time ...

• Read the following papers
  – “Low-Cost Supervision for Multiple-Source Attribute Extraction”, Joseph Reisinger and Marius Pasca
  – “Weakly-Supervised Acquisition of Open-Domain Classes and Class Attributes from Web Documents and Query Logs”, Marius Pasca and Benjamin Van Durme

• Answer the following questions on a piece of paper
  – Define what is an attribute
  – Describe what is the attribute extraction task
  – Why is it important
  – What kinds of application would benefit from it
  – Describe the approach and give an example
  – Described the evaluation
  – Discuss +/- of the approach
  – Propose an alternative solution to the problem