Deep Neural Networks for Information Extraction

Tomáš Gogár, Petr Baudiš
Information Extraction

Data prepared for computer analysis
Information extraction: EASY

- Databases
Information Extraction

Data partially prepared for computer analysis

Information extraction: EASY

- Markup documents
- Databases
Information Extraction

Data NOT prepared for computer analysis
Information extraction: DIFFICULT

- Semistructured documents
- Markup documents
- Databases

Complexity of Information Extraction vs. Structuredness
Information Extraction

- Plain Text
- Semistructured documents
- Markup documents
- Databases

Data NOT prepared for computer analysis

Information extraction: VERY DIFFICULT
Information Extraction

Structuredness

Complexity of Information Extraction

- Plain Text
- Petr Baudiš
- Semistructured documents
- Tomáš Gogár
- Markup documents
- Databases
Web Information Extraction - Current systems

- Web pages are created from Templates
- Learn template structure $\Rightarrow$ Extract Information
Web Information Extraction - Current systems

- Web pages are created from Templates
- Learn template structure $\Rightarrow$ Extract Information
- Template learning:
  - Manual annotation - Scraping
Web Information Extraction - Current systems

- Web pages are created from Templates
- Learn template structure $\Rightarrow$ Extract Information
- Template learning:
  - Manual annotation
  - Automatic learning - repeated patterns
Web Information Extraction - Current systems

- Web pages are created from Templates
- **Learn template structure** ⇒ **Extract Information**
- **Template learning:**
  - Manual annotation
  - Automatic learning - repeated patterns

It’s just a hack!
What matters in Information Extraction

What is written?

Where it is written?

How it is written?
What matters in Information Extraction

**What** is written?

**Where** it is written?

**How** it is written?  

[Screenshot]
What matters in Information Extraction

**What** is written?

**Where** it is written?

**How** it is written?

[SPATIAL BAG-OF-WORDS]

[Screenshot]
Intro: Bag-of-Words

Text representation often used in NLP:

(1) John likes to watch movies. Mary likes movies too.

(2) John also likes to watch football games.

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;John&quot;</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;likes&quot;</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;to&quot;</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;watch&quot;</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;movies&quot;</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;also&quot;</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;football&quot;</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;games&quot;</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;Mary&quot;</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;too&quot;</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Vectors:

Document1 = [1, 2, 1, 1, 2, 0, 0, 0, 1, 1]
Document2 = [1, 1, 1, 1, 0, 1, 1, 1, 0, 0]
Intro: Hashing Trick

Text representation often used in NLP:

(1) John likes to watch movies. Mary likes movies too.

(2) John also likes to watch football games.

Vocabulary:

- Hashing function
  \[ h(\text{john}) = 3 \]
  \[ h(\text{likes}) = 1 \]
  ...

- Does not need vocabulary :-)
- Arbitrary size of result vector :-)
- Collisions :-(
Spatial Bag-of-Words

- We do not process text as a whole
- We process each TEXT NODE individually
Spatial Bag-of-Words

- We do not process text as a whole
- We process each TEXT NODE individually
Spatial Bag-of-Words

- We do not process text as a whole
- We process each TEXT NODE individually
Spatial Bag-of-Words

- We do not process text as a whole
- We process each TEXT NODE individually
Spatial Bag-of-Words

- We do not process text as a whole
- We process each TEXT NODE individually

TEXT ENCODED IN TENSOR
(SAME AS IMAGE)
Net architecture

Problem: Uses only local information!
Spatial likelihood
Final system

- Image of Page
- Spatially Encoded Text (TextMaps)
- Interpreted Web Page
- Positions of candidates: [120, 32, 140, 55], [30, 500, 200, 575]
- Neural net (Web Page features)
- Spatial likelihood
- argmax
  - $P(\text{class} \mid \text{candidate}_\text{context})$: [0.92, 0.23], ...
  - $P(\text{class} \mid \text{candidate}_\text{position})$: [0.65, 0.24], ...

Diagram showing the final system, where a combination of models and data sources (image, text, positions) are integrated through spatial and neural network models to determine a final output through an argmax process.
## Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Image Accuracy</th>
<th>Price Accuracy</th>
<th>Name accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeuralNet + Spatial.</td>
<td>98.7±1.6</td>
<td>95.3±6.6</td>
<td>87.1±15.0</td>
</tr>
<tr>
<td>NeuralNet</td>
<td>95.9±2.9</td>
<td>86.2±9.3</td>
<td>78.4±19.0</td>
</tr>
<tr>
<td>Baseline: Heuristic + Spatial.</td>
<td>63.7±20.1</td>
<td>73.6±18.8</td>
<td>34.4±20.5</td>
</tr>
<tr>
<td>Baseline: Spatial</td>
<td>46.5±18.7</td>
<td>9.7±14.4</td>
<td>12.2±12.0</td>
</tr>
</tbody>
</table>

Table 3. Comparison of algorithms: mean and standard deviation of accuracy across 10 splits (in percents).

<table>
<thead>
<tr>
<th>Neural net inputs</th>
<th>Image Accuracy</th>
<th>Price Accuracy</th>
<th>Name accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screenshot + TextMap</td>
<td>95.9±2.9</td>
<td>86.2±9.3</td>
<td>78.4±19.0</td>
</tr>
<tr>
<td>Screenshot</td>
<td>93.5±7.4</td>
<td>73.3±19.4</td>
<td>73.4±16.0</td>
</tr>
<tr>
<td>TextMap</td>
<td>41.4±18.6</td>
<td>77.0±17.9</td>
<td>49.4±18.0</td>
</tr>
</tbody>
</table>

Table 4. Neural Net with different input data: mean and standard deviation of accuracy across 10 splits (in percents).
Results

**Fig. 5.** Examples of *current price* detection.
Results

Fig. 6. Examples of product names divided into two parts (manufacturer + model).
Future work

Machine learning:

- Solve global position problem: Attention network?
- Try to learn text features
- Try other similar tasks: ex. classification

Practical problems:

- Popup windows
- Information distributed in multiple DOM-Elements
Source code:
github.com/gogartom/TextMaps