Introduction

Extraction
- Pattern-based approach
- Distributional approach

Evaluation
Automatic relation extraction
Semantic Networks

- network representing *relations between concepts*
- *knowledge graph*
- WordNet – lexical database of English
  - synsets, main relation hyponymy/hypernymy, meronymy, synonymy, antonymy…
  - Multilingual Wordnet network
Why would you do that?

- semantic analysis (house → home, music, MD?)
- query expansion (dog → poodle, terrier...)
- lexical substitution (match → game)
- machine translation
- question answering
- domain classification (lemon, apple, banana → fruit)
- summarization
- paraphrase
Why would you do that?

- semantic analysis (house → home, music, MD?)
- query expansion (dog → poodle, terrier . . .)
- lexical substitution (match → game)
- machine translation
- question answering
- domain classification (lemon, apple, banana → fruit)
- summarization
- paraphrase

Example

Human illuminates Document
AG[bird:1] VERB sezobnout SUBS[feed:1]
What do we need?

- morphological tags
- syntactic analysis (phrases)
- dataset (dictionary, corpus, Wikipedia...)
Pattern recognition

regular expression to match Part-of-Speech and text
Pattern recognition

regular expression to match Part-of-Speech and text

Example

NP {,} especially {NP, }* {or |and} NP

...most European countries, especially France, England, and Spain.

...e.g. apples, bananas, or pears.
Pattern recognition

regular expression to match Part-of-Speech and text

Example

```plaintext
NP {,} especially {NP, }* {or |and} NP
...most European countries, especially France, England, and Spain.
European country > France
European country > England
European country > Spain
```
Pattern recognition

regular expression to match Part-of-Speech and text

**Example**

```
NP {,,} especially {NP, }* {or |and} NP
...most *European countries*, especially *France, England, and Spain*.
European country > France
European country > England
European country > Spain
```

**Example**

e.g. {NP,}* {and |or} NP.
Pattern recognition

regular expression to match Part-of-Speech and text

Example

NP {,} especially {NP,} * {or | and} NP
...most European countries, especially France, England, and Spain.
European country > France
European country > England
European country > Spain

Example

e.g. {NP,} * {and | or} NP.
...e.g. apples, bananas, or pears.
related terms
Example

NP such as \{NP, \} \ast \{\text{and} \mid \text{or}\} \text{NP}
Example

NP such as \{NP, \}* \{and | or\} NP

common *domestic animals* such as the *ferret* and the *fancy rat*

domestic animal \(\rightarrow\) ferret

domestic animal \(\rightarrow\) (fancy) rat
Example

NP such as \{NP, \} \ast \{\text{and } | \text{or}\} \text{ NP}

common *domestic animals* such as the *ferret* and the *fancy rat*

domestic animal $\rightarrow$ ferret

domestic animal $\rightarrow$ (fancy) rat

in areas with a long history of *mining* such as *South-west England*

mining $\rightarrow$ South-west England
Example

NP such as \{NP, \} \ast \{\text{and} \mid \text{or}\} \ NP

common *domestic animals* such as the *ferret* and the *fancy rat*

domestic animal >ferret

domestic animal >(fancy) rat

in areas with a long history of *mining* such as *South-west England*

mining >South-west England

in *areas* (with a long history of mining) such as *South-west England*

area >South-west England

- remove stopwords
- detect optional adjunct phrases
- detect named entities
<table>
<thead>
<tr>
<th>No.</th>
<th>Pattern</th>
<th>Number of occurrences</th>
<th>Number of relevant occurrences</th>
<th>Intermediary precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>other than</td>
<td>168</td>
<td>164</td>
<td>97.6</td>
</tr>
<tr>
<td>2</td>
<td>especially</td>
<td>120</td>
<td>90</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>principally</td>
<td>11</td>
<td>6</td>
<td>54.5</td>
</tr>
<tr>
<td>4</td>
<td>usually</td>
<td>18</td>
<td>14</td>
<td>77.8</td>
</tr>
<tr>
<td>5</td>
<td>such as</td>
<td>2470</td>
<td>1950</td>
<td>78.9</td>
</tr>
<tr>
<td>6</td>
<td>in particular</td>
<td>78</td>
<td>48</td>
<td>61.5</td>
</tr>
<tr>
<td>7</td>
<td>e(.).g(.)</td>
<td>280</td>
<td>216</td>
<td>77.1</td>
</tr>
<tr>
<td>8</td>
<td>become</td>
<td>780</td>
<td>510</td>
<td>66.7</td>
</tr>
<tr>
<td>9</td>
<td>another</td>
<td>92</td>
<td>72</td>
<td>78.3</td>
</tr>
<tr>
<td>10</td>
<td>notably</td>
<td>76</td>
<td>42</td>
<td>55.3</td>
</tr>
<tr>
<td>11</td>
<td>particularly</td>
<td>130</td>
<td>80</td>
<td>61.5</td>
</tr>
<tr>
<td>12</td>
<td>except</td>
<td>13</td>
<td>4</td>
<td>30.8</td>
</tr>
<tr>
<td>13</td>
<td>called</td>
<td>270</td>
<td>220</td>
<td>81.5</td>
</tr>
<tr>
<td>14</td>
<td>like</td>
<td>1600</td>
<td>1300</td>
<td>81.3</td>
</tr>
<tr>
<td>15</td>
<td>including</td>
<td>670</td>
<td>430</td>
<td>64.2</td>
</tr>
</tbody>
</table>
Corpus query

- special case of pattern recognition, CQL query
- bigger data at hand, less options
Corpus query

- special case of pattern recognition, CQL query
- bigger data at hand, less options

Example

`je/jsou`

2: `[k="k1"&c="c1"] ([lc="","] [k="k1"])*
   ([lc="a"|lc="i"|lc="nebo"|lc="či"] [k="k1"])?
   [lemma_lc="být"&tag="k5eAaImIp3.*"&lc!="ne.*"]
   ([k="k1"&c="c[1246]" [k="k2"]{0,2})?
1: `[k="k1"&c="c[1246]"]`

experiment on domain dictionary: precision 40 %, when limited to
dictionary terms 52 %
Multilingual translation

using translation equivalents from multilingual dictionary to provide synonyms
Multilingual translation

using translation equivalents from multilingual dictionary to provide synonyms

<table>
<thead>
<tr>
<th>Example</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>stůl = table</td>
<td></td>
</tr>
<tr>
<td>table = stůl, stolek</td>
<td></td>
</tr>
<tr>
<td>stůl = stolek</td>
<td></td>
</tr>
</tbody>
</table>
Synonym transitivity

- expanding relations based on existing relations (transitive closure)

Example

city = town, town = municipality
⇒ city = municipality
Distributional approach

- vector space model
- word-context frequency matrix
- clustering
- similar context $\neq$ synonym
- e.g. Sketch Engine thesaurus
TOEFL test evaluation

- evaluation by solving TOEFL synonym test
- Choose synonym for *fabricate*.
  - construct, alter, select, demonstrate
- build synonym set for each word
- detect overlap
- success rate 88 %
SemEval

- various tasks evaluating computational semantic analysis systems
- human annotators provide *gold standards*
- NLP systems are evaluated
- tasks include Word Sense Disambiguation, Machine Translation, Information Extraction, Learning Semantic Relations...
References I

Hyponymy patterns: Semi-automatic extraction, evaluation and inter-lingual comparison.
In *Text, Speech and Dialogue*, pages 37–44.

Inriasac: Simple hypernym extraction methods.

Automated discovery of wordnet relations.
Evaluation of automatic hypernym extraction from technical corpora in english and dutch.

Lexical patterns or dependency patterns: which is better for hypernym extraction?

A combined pattern-based and distributional approach for automatic hypernym detection in dutch.
In RANLP, pages 593–600.