Semantic Applications of Text Processing

LUCAS-dagen

Pierre Nugues

Lunds Tekniska Högskola
Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

October 21, 2010
A first observation: text is the largest repository of human knowledge. The most popular web applications today are text applications. Many ways to model, extract, and process knowledge in text. Semantics, the holy grail of text understanding?

Presentation in three parts:

- Information extraction
- Application example: Carsim
- Semantic parsing
Text processing is everywhere now:

- Interface display and localization
- Spelling and grammatical checkers: Microsoft Word
- Search: Google, Bing, company web sites.
- Information extraction
- Telephone servers
- Translation: Google Translate, Microsoft/Bing Translator
Unicode is an attempt to represent most alphabets. Not only an encoding framework, but also a collation algorithm, rules to present dates, time, and numbers.

From *Programming Perl* by Larry Wall, Tom Christiansen, Jon Orwant, O’Reilly, 2000:

> If you don’t know yet what Unicode is, you will soon—even if you skip reading this chapter—because working with Unicode is becoming a necessity.

Perl’s reach is probably tied to the advent of the web. Perl made text processing easy (regular expressions). Unicode was a necessity because (multilingual) text processing is a necessity.
Information extraction: One of the first agnostic semantic application. Started with the **Message understanding conferences** (MUC), a benchmarking competition organized by the US military (1987–1997). The first task of the MUCs is the extraction of names (proper nouns), time expressions, and money quantities.

[PERSON Wolff], currently a journalist in [LOCATION Argentina], played with [PERSON Del Bosque] in the final years of the seventies in [ORGANIZATION Real Madrid].

Often referred to as **named entity recognition** (NER).
The most challenging task of MUCs is referred to as information extraction. It consists of:

- The analysis of pieces of text ranging from one to two pages,
- The identification of entities or events of a specified type,
- The filling of a pre-defined template with relevant information from the text.

Information extraction then transforms free texts into tabulated information: a semantic representation.
San Salvador, 19 Apr 89 (ACAN-EFE) – [TEXT] Salvadoran President-elect Alfredo Cristiani condemned the terrorist killing of Attorney General Roberto Garcia Alvarado and accused the Farabundo Marti National Liberation Front (FMLN) of the crime...

Garcia Alvarado, 56, was killed when a bomb placed by urban guerrillas on his vehicle exploded as it came to a halt at an intersection in downtown San Salvador...
<table>
<thead>
<tr>
<th>Template slots</th>
<th>Information extracted from the text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident: Date</td>
<td>19 Apr 89</td>
</tr>
<tr>
<td>Incident: Location</td>
<td>El Salvador: San Salvador (city)</td>
</tr>
<tr>
<td>Incident: Type</td>
<td>Bombing</td>
</tr>
<tr>
<td>Perpetrator: Individual ID</td>
<td>urban guerrillas</td>
</tr>
<tr>
<td>Perpetrator: Organization ID</td>
<td>FMLN</td>
</tr>
<tr>
<td>Perpetrator: Organization confidence</td>
<td>Suspected or accused by authorities: FMLN</td>
</tr>
<tr>
<td>Physical target: Description</td>
<td>vehicle</td>
</tr>
<tr>
<td>Physical target: Effect</td>
<td>Some damage: vehicle</td>
</tr>
<tr>
<td>Human target: Name</td>
<td>Roberto Garcia Alvarado</td>
</tr>
<tr>
<td>Human target: Description</td>
<td>Attorney general: Roberto Garcia Alvarado</td>
</tr>
<tr>
<td></td>
<td>driver</td>
</tr>
<tr>
<td></td>
<td>bodyguards</td>
</tr>
<tr>
<td>Human target: Effect</td>
<td>Death: Roberto Garcia Alvarado</td>
</tr>
<tr>
<td></td>
<td>No injury: driver</td>
</tr>
<tr>
<td></td>
<td>Injury: bodyguards</td>
</tr>
</tbody>
</table>
FASTUS’ Architecture

The FASTUS system was designed at the Stanford Research Institute to extract information from free-running text. FASTUS uses partial parsers organized as a cascade of finite-state automata.

---

**Figure 9.2.** A cascade of partial parsers.

- **Sentences** → **Tokenizer** → **Multiwords** → **Part-of-speech tagging** → **Group detection (or chunking)**

---

9.8 Elementary Analysis of Grammatical Functions

9.8.1 Main Functions

In a previous section, we named groups according to the part of speech of their main word, that is, noun groups and verb groups. We can also consider their grammatical function in the sentence. We already saw that main functions (or relations) are subject, direct object, and indirect object. An accurate detection of function is difficult, but we can write a simplified one using cascaded parsing and phrase-structure rules.
Demonstrations

- OpenCalais: http://viewer.opencalais.com/
- European media monitor: http://press.jrc.it/geo?type=event&format=html&language=all
- Carsim: http://nlp.cs.lth.se/
Some definitions:

1. The automatic extraction of predicate–argument structures in sentences and clauses.
2. Determine events, their actors (roles) and circumstances. Tesnière used the word drama to refer to the events.
3. Generic component to applications such as information extraction.
In FrameNet, **Revenge** as an example of predicate (frame), which features five roles (frame elements): **Avenger**, **Punishment**, **Offender**, **Injury**, and **Injured_party**.

1. $<$Avenger$>$ His brothers] **avenged** $<$Injured_party$>$ him].
2. With this, $<$Avenger$>$ El Cid] at once **avenged** $<$Injury$>$ the death of his son].
3. $<$Avenger$>$ Hook] tries to **avenge** $<$Injured_party$>$ himself] $<$Offender$>$ on Peter Pan] $<$Punishment$>$ by becoming a second and better father].
Significance

- Invariant across different syntactic realizations:
  - Pierre gave a presentation on semantic applications on October 21
  - On October 21, Pierre gave a presentation on semantic applications
  - A presentation on semantic applications was given by Pierre on October 21

- Semantic role labeling applications:
  - Question answering
  - Information extraction
  - Document categorization
  - Machine translation
  - Speech recognition

Parsing sentence required 34ms.
Events, either verbs or nouns, are represented as *predicates*.

Classes of predicates define sets of participants, the *roles*.

For example in Propbank, **have.03** has two roles:

- **Arg0**: owner
- **Arg1**: possession

Participants (roles) and circumstances (adjuncts) are the *arguments*.

Relation to predicate logic, e.g. in Prolog:

> 'have.03'('They', brandy, 'in the library')

$\text{–Root–} \rightarrow \text{They} \quad \text{had} \quad \text{brandy} \quad \text{in the library}$. 

$\text{have.03}$

(A0)

(A1)

AM–LOC
LTH Semantic Parser

They had brandy in the library.

Predicate Disambiguation (PD)

They had brandy in the library.

Argument Identification (AI)

They had brandy in the library.

Argument Classification (AC)

They had brandy in the library.
## Parser’s Performance

<table>
<thead>
<tr>
<th>Rank</th>
<th>Rank in task</th>
<th>System</th>
<th>Average</th>
<th>Catalan</th>
<th>Chinese</th>
<th>Czech</th>
<th>English</th>
<th>German</th>
<th>Japanese</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 (SRLonly)</td>
<td>Zhao</td>
<td>80.47</td>
<td>@ 80.32</td>
<td>77.72</td>
<td>85.19</td>
<td>85.44</td>
<td>75.99</td>
<td>78.15</td>
<td>@ 80.46</td>
</tr>
<tr>
<td>2</td>
<td>2 (SRLonly)</td>
<td>Nugues</td>
<td>80.31</td>
<td>80.01</td>
<td>@ 78.60</td>
<td>85.41</td>
<td>85.63</td>
<td>@ 79.71</td>
<td>76.30</td>
<td>76.52</td>
</tr>
<tr>
<td>3</td>
<td>1 (Joint)</td>
<td>Chen</td>
<td>79.96</td>
<td>80.10</td>
<td>76.77</td>
<td>82.04</td>
<td>@ 86.15</td>
<td>76.19</td>
<td>78.17</td>
<td>80.29</td>
</tr>
<tr>
<td>4</td>
<td>2 (Joint)</td>
<td>Che</td>
<td>79.94</td>
<td>77.10</td>
<td>77.15</td>
<td>@ 86.51</td>
<td>85.51</td>
<td>78.61</td>
<td>@ 78.26</td>
<td>76.47</td>
</tr>
<tr>
<td>5</td>
<td>3 (Joint)</td>
<td>Merlo</td>
<td>78.42</td>
<td>77.44</td>
<td>76.05</td>
<td>86.02</td>
<td>83.24</td>
<td>71.78</td>
<td>77.23</td>
<td>77.19</td>
</tr>
<tr>
<td>6</td>
<td>3 (SRLonly)</td>
<td>Meza-Ruiz</td>
<td>77.46</td>
<td>78.00</td>
<td>77.73</td>
<td>75.75</td>
<td>83.34</td>
<td>73.52</td>
<td>76.00</td>
<td>77.91</td>
</tr>
<tr>
<td>7</td>
<td>4 (Joint)</td>
<td>Bohnet</td>
<td>76.00</td>
<td>74.53</td>
<td>75.29</td>
<td>79.02</td>
<td>80.39</td>
<td>75.72</td>
<td>72.76</td>
<td>74.31</td>
</tr>
<tr>
<td>8</td>
<td>5 (Joint)</td>
<td>Asahara</td>
<td>75.65</td>
<td>72.35</td>
<td>74.17</td>
<td>84.69</td>
<td>84.26</td>
<td>63.66</td>
<td>77.93</td>
<td>72.50</td>
</tr>
<tr>
<td>9</td>
<td>6 (Joint)</td>
<td>Brown</td>
<td>72.85</td>
<td>72.18</td>
<td>72.43</td>
<td>78.02</td>
<td>80.43</td>
<td>73.40</td>
<td>61.57</td>
<td>71.95</td>
</tr>
<tr>
<td>10</td>
<td>7 (Joint)</td>
<td>Dai</td>
<td>70.78</td>
<td>66.34</td>
<td>71.57</td>
<td>75.50</td>
<td>78.93</td>
<td>67.43</td>
<td>71.02</td>
<td>64.64</td>
</tr>
<tr>
<td>11</td>
<td>8 (Joint)</td>
<td>Zhang</td>
<td>70.31</td>
<td>67.34</td>
<td>73.20</td>
<td>78.28</td>
<td>77.85</td>
<td>62.95</td>
<td>64.71</td>
<td>67.81</td>
</tr>
<tr>
<td>12</td>
<td>9 (Joint)</td>
<td>Lu Li</td>
<td>69.72</td>
<td>66.95</td>
<td>67.06</td>
<td>79.08</td>
<td>77.17</td>
<td>61.98</td>
<td>69.58</td>
<td>66.23</td>
</tr>
<tr>
<td>13</td>
<td>4 (SRLonly)</td>
<td>Baoli Li</td>
<td>69.26</td>
<td>74.06</td>
<td>70.37</td>
<td>57.46</td>
<td>69.63</td>
<td>67.76</td>
<td>72.03</td>
<td>73.54</td>
</tr>
<tr>
<td>14</td>
<td>10 (Joint)</td>
<td>Vallejo</td>
<td>68.95</td>
<td>70.14</td>
<td>66.71</td>
<td>71.49</td>
<td>75.97</td>
<td>61.01</td>
<td>68.82</td>
<td>68.48</td>
</tr>
<tr>
<td>15</td>
<td>5 (SRLonly)</td>
<td>Moreau</td>
<td>66.49</td>
<td>65.60</td>
<td>67.37</td>
<td>71.74</td>
<td>72.14</td>
<td>66.50</td>
<td>57.75</td>
<td>64.33</td>
</tr>
<tr>
<td>16</td>
<td>11 (Joint)</td>
<td>Lluís</td>
<td>63.06</td>
<td>46.79</td>
<td>59.72</td>
<td>76.90</td>
<td>75.86</td>
<td>62.66</td>
<td>71.60</td>
<td>47.88</td>
</tr>
<tr>
<td>17</td>
<td>6 (SRLonly)</td>
<td>Täckström</td>
<td>61.27</td>
<td>57.11</td>
<td>63.41</td>
<td>71.05</td>
<td>67.64</td>
<td>53.42</td>
<td>54.74</td>
<td>61.51</td>
</tr>
<tr>
<td>18</td>
<td>7 (SRLonly)</td>
<td>Lin</td>
<td>57.18</td>
<td>61.70</td>
<td>70.33</td>
<td>60.43</td>
<td>65.66</td>
<td>59.51</td>
<td>23.78</td>
<td>58.87</td>
</tr>
<tr>
<td>19</td>
<td>12 (Joint)</td>
<td>Ren</td>
<td>56.69</td>
<td>41.00</td>
<td>72.58</td>
<td>62.82</td>
<td>67.56</td>
<td>54.31</td>
<td>58.73</td>
<td>39.80</td>
</tr>
<tr>
<td>20</td>
<td>13 (Joint)</td>
<td>Zeman</td>
<td>32.14</td>
<td>24.19</td>
<td>34.71</td>
<td>58.13</td>
<td>36.05</td>
<td>16.44</td>
<td>30.13</td>
<td>25.36</td>
</tr>
</tbody>
</table>
LTH Parser: http://barbar.cs.lth.se:8081/
Code available from Google code:
http://code.google.com/p/mate-tools/
TextRunner:
http://www.cs.washington.edu/research/textrunner/