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

Semantics

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

Language Technology

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).

Information Extraction

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.

An Example

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...

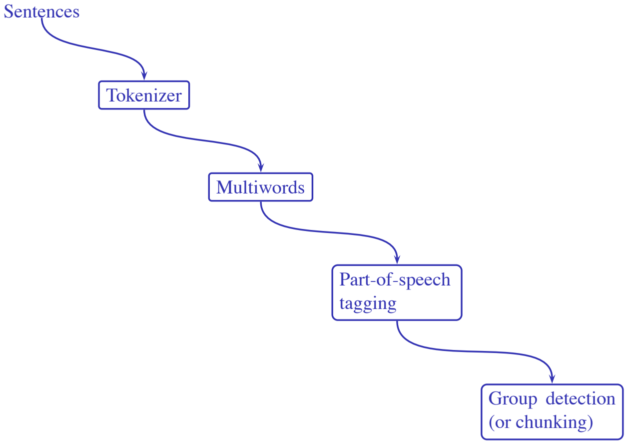
The Template

Template slots	Information extracted from the text
Incident: Date	19 Apr 89
Incident: Location	El Salvador: San Salvador (city)
Incident: Type	Bombing
Perpetrator: Individual ID	*urban guerrillas*
Perpetrator: Organization ID	*FMLN*
Perpetrator: Organization confidence	Suspected or accused by authorities: *FMLN*
Physical target: Description	*vehicle*
Physical target: Effect	Some damage: *vehicle*
Human target: Name	*Roberto Garcia Alvarado*
Human target: Description	*Attorney general*: *Roberto Garcia Alvarado* *driver* *bodyguards*
Human target: Effect	Death: *Roberto Garcia Alvarado* No injury: *driver* Injury: *bodyguards*

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.



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.
4. Semantic parsing needs a dictionary: FrameNet, ProbPank, VerbNet (`http://verbs.colorado.edu/verb-index/`).

Semantic Parsing: An Example

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].

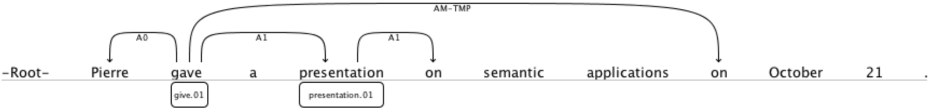
Significance

- 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

Try to answer questions: *who? whom? what? when? where?*

	Pierre	gave	a	presentation	on	semantic	applications	on	October	21	.
give.01	A0		A1					AM-TMP			
presentation.01					A1						

Parsing sentence required 34ms.

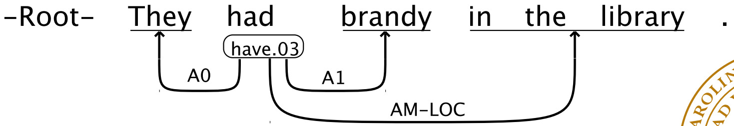


Semantic Dependencies

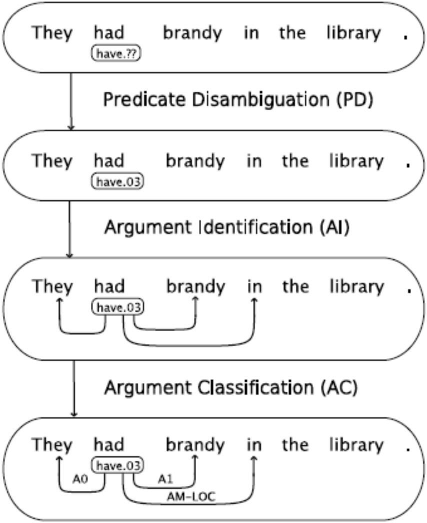
- Events, either verbs or nouns, are represented as *predicates*
- Classes of predicates define sets of participants, the *roles*
  For example in Propbank, **have.03** has two roles:

  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')`



LTH Semantic Parser



Parser's Performance

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

Demonstrations

- 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/`