Extracting subcategorisation information from the LADL-tables

Claire Gardent (CNRS/LORIA, Nancy)
Bruno Guillaume (INRIA/LORIA, Nancy)
Guy Perrier (Université de Nancy 2, Nancy)
Ingrid Falk (CNRS/LORIA, Nancy)

13 March2006

Overview

- The LADL Tables
- Subcategorisation information in NLP
- Existing subcategorisation lexicons for French
- Converting the LADL tables into a subcategorisation lexicon
- Going further: validation and fusion with other lexicons

The LADL Tables

- Maurice Gross's **Grammar lexicon** is a very large scale, high precision linguistic resource developed over several years by a group of skilled linguists and according to well defined linguistic criteria
- It was encoded into a digital format in the **LADL** tables

Grammar and lexicon

- Grammars/Syntactic theories define general rules describing the syntagmatic structures of sentences
- But there are many exceptions
  e.g., not all transitive verbs take the passive form (*to weigh*)
- Thus a complete description of a language must include both generalisations and lexical constraints on these generalisations

$\Rightarrow$ The grammar lexicon lists these constraints for predicative items (verbs, predicative nouns, predicative adjectives, support verb constructions)

Maurice Gross' Grammar-Lexicon

- Describes the syntactico-semantic properties of (French) basic sentences
- Consists of a set of Tables (the **LADL tables**)
- A **table** gathers together predicative items (verbs, support adjectival/nominal verb constructions) with comparable syntactico-semantic behaviour
- In a table, **columns** further specifies the syntactico-semantic properties of each verb in that table

Example table

Table 8 Description de la table

N0 =: Nhum	N0 =: Nnr	N0 =: le fait Qu P	N0 = V1 W	[extrap]		8	N0 est V-ant	N0 V	N0 est Vpp W	N1 =: Qu P	N1 =: Qu Psubj	[pc z.]	N1 =: si P ou si P	N1 = V0 W	Tc =: futur	Tc =: passé	Vc =: devoir	Vc =: pouvoir	Vc =: savoir	N1 =: ce(ci+la)	N1 =: Ppv	de N1 = de là	N1 =: Nhum	N1 =: N-hum	N1 =: le fait Qu P	Prép Nhum = Ppv	[extrap][passif]	de N1 V N0	N0 V contre Nhum
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
+	-	-	-	-	s'	abstenir	+	+	-	-	+	-	-	+	-	-	-	-	-	+	+	-	-	+	-	+	-	-	-
+	-	-	-	-		abuser	-	+	-	+	+	-	-	+	+	-	-	+	+	+	+	-	+	+	+	+	+	-	-

Subcategorisation information in the Grammar-Lexicon

- Detailed
- Large coverage

Detailed

- Each verb in the GL may be associated with **several verb usages**
- Each table associates with all its entries one (sometime two) **basic subcategorisation frame** (e.g., n0V)
- For each entry in a table, the columns of the table further specifies the syntactico-semantic properties of that entry (verb) and in particular
  - **further subcategorisation frames** that might be valid for that verb
  - **detailed information about the verb and the arguments** appearing in the various subcategorisation frames

Information about verb and arguments

- for the verb: verb type, auxiliary used, temporal agreement constraints between verb an sentential complt, etc.
- for nominal arguments : constraints on animacy, number; selectional restrictions; pronominalisation; restriction on the determiner; etc.
- for prepositional arguments: information about type and value of the preposition, about the thematic role fulfilled, etc.

Large scale

- 6 500 verbs (7 357 in Morphalou, 5 381 in LEFFF)
- 31 000 entries distributed over 81 tables
  - 20 0000 collocations (in 20 tables)
  - 3 000 verbs with sentential complements (in 18 tables)
  - 8 000 verbs with nominal complements (43 tables)
- 5 000 nouns
- 3 000 adjectives

Subcategorisation information in NLP

- (Briscoe and Carroll, 1993): Half of parse failures results from inaccurate subcategorisation information
- (Carroll and Fang, 2004): enriching an HPSG grammar with detailed subcategorisation information improves the parse success rate by 15%
- (Han et al. 2000): subcategorisation information is a key factor in achieving good quality machine translation
- (Jijkoun et al 2004): extracting syntactic relations between entities substantially increases the performance of a question answering system

Syntactic Lexicons for NLP

- Each verb usage is associated with the set of subcategorisation frames associated with that verb by the GL
- A **subcategorisation frame** is a list of feature structures where each feature structure encodes properties of the arguments or of the verb

Existing computational syntactic lexicons for English

Syntactic lexicons lists for each verb usage the subcategorisation frames admitted by that verb usage e.g.,

- COMLEX Syntax (Macleod et al. 1994): 6 000 verbs
- VerbNet (Palmer et al. 2000): 4 000 verb senses, 52 subcategorisation frames

Existing computational syntactic lexicons for French

- Proton, `http://bach.arts.kuleuven.be/PA/proton.html`
  3700 verbs and 8 600 entries
  Non standard format, not directly usable for NLP
- LEFFF, `http://www.labri.fr/perso/clement/lefff/`
  5 381 verbs, freely avalaible, NLP lexicon
  Obtained through statistical methods, never evaluated

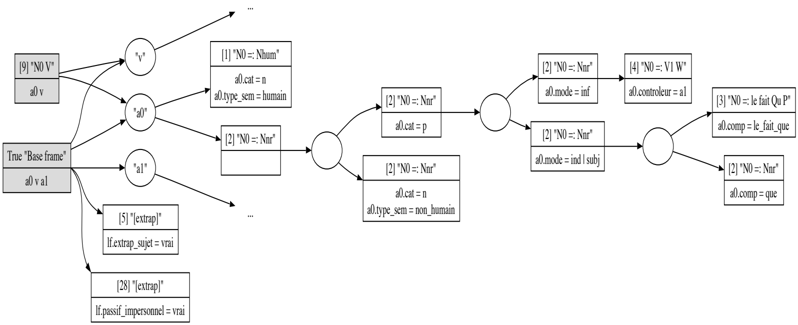
Goal: use the LADL tables as a way to validate and complement LEFFF, PROTON or/and syntactic lexicon acquired from corpora (Polonium cooperation)

Extracting a syntactic lexicon from the LADL tables

Method:

- Specify an and-or graph representing the content of the table
- Process this graph to produce a computational subcategorisation lexicon

Example graph



Example output

```
--abuser--
a0[cat=n, type_sem=humain] v[...]
  lf[passif_impersonnel=vrai]
a0[cat=n, type_sem=humain] v[...] a1[...]
  lf[passif_impersonnel=vrai]

--beneficier--
a0[cat=n, type_sem=non_humain] v[...] a1[...]
  lf[extrap_sujet=vrai, passif_impersonnel=vrai]
a0[cat=p, mode=inf] v[...] a1[...]
  lf[extrap_sujet=vrai, passif_impersonnel=vrai]
a0[cat=p, mode=ind|subj, comp=le_fait_que] v[...] a1[...]
  lf[extrap_sujet=vrai, passif_impersonnel=vrai]
a0[cat=p, mode=ind|subj, comp=que] v[...] a1[...]
  lf[extrap_sujet=vrai, passif_impersonnel=vrai]
```

Why an and-or graph?

- The graph makes explicit the structure of the tables and in particular:
  - **Conjunctions** (AND-nodes)
  - **Disjunctions** (OR-nodes)
  - **Dependencies** (Graph EDGES)
  - **Feature-Structure information** (Graph NODES)
- The graph provides both a declarative and a procedural interpretation of the tables

Example (Table 8)

Table 8 Description de la table

N0 =: Nhum	N0 =: Nnr	N0 =: le fait Qu P	N0 = V1 W	[extrap]		8	N0 est V-ant	N0 V	N0 est Vpp W	N1 =: Qu P	N1 =: Qu Psubj	[pc z.]	N1 =: si P ou si P	N1 = V0 W	Tc =: futur	Tc =: passé	Vc =: devoir	Vc =: pouvoir	Vc =: savoir	N1 =: ce(ci+la)	N1 =: Ppv	de N1 = de là	N1 =: Nhum	N1 =: N-hum	N1 =: le fait Qu P	Prép Nhum = Ppv	[extrap][passif]	de N1 V N0	N0 V contre Nhum
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
+	-	-	-	-	s'	abstenir	+	+	-	-	+	-	-	+	-	-	-	-	-	+	+	-	-	+	-	+	-	-	-
+	-	-	-	-		abuser	-	+	-	+	+	-	-	+	+	-	-	+	+	+	+	-	+	+	+	+	+	-	-

- Colums 13 and 14 **depend** on column 11 and 12: a verb will accept an infinitivalor an interrogative complement only if it accepts a sentential complement.
- Columns 16 and 17 specify **disjunctive** information: the infinitival complement is compatible with a past tense adverbial, a future tense adverbial, both or neither.

Example (Table 8)

Table 8 Description de la table

N0 =: Nhum	N0 =: Nnr	N0 =: le fait Qu P	N0 = V1 W	[extrap]		8	N0 est V-ant	N0 V	N0 est Vpp W	N1 =: Qu P	N1 =: Qu Psubj	[pc z.]	N1 =: si P ou si P	N1 = V0 W	Tc =: futur	Tc =: passé	Vc =: devoir	Vc =: pouvoir	Vc =: savoir	N1 =: ce(ci+la)	N1 =: Ppv	de N1 = de là	N1 =: Nhum	N1 =: N-hum	N1 =: le fait Qu P	Prép Nhum = Ppv	[extrap][passif]	de N1 V N0	N0 V contre Nhum
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
+	-	-	-	-	s'	abstenir	+	+	-	-	+	-	-	+	-	-	-	-	-	+	+	-	-	+	-	+	-	-	-
+	-	-	-	-		abuser	-	+	-	+	+	-	-	+	+	-	-	+	+	+	+	-	+	+	+	+	+	-	-

- Column 2 specifies **disjunctive** information on the argument realisation: the subject is unrestricted i.E. can be an NP, an infinitival or a finite sentence.
- Colums 6 and 7 specify a **conjunctive** information about the verb (its lemma and its reflexivity)

Graph syntax

Or-nodes : ellipses; indicate a disjunction

And-nodes : rectangles; indicate a conjunction; 2 parts:

- Top part : condition
  - `[`c`]` is `True` if column c contains + or is non empty (for a lemma giving column). Else `False`.
  - `[!`c`]` is `True` if column c contains – or is empty. Else `False`.
- Bottom part : Feature structure specification
  - $arg.feat = value$ where *arg* can be `v`, `a0`, `a1`, `a2` or `lf`
  - A feature value *value* is either a disjunction of atomic values or the symbol $\$c$ which indicates the vlaue given by the content of the cell $[l,c]$ with l and c a table lign and column number respectively.

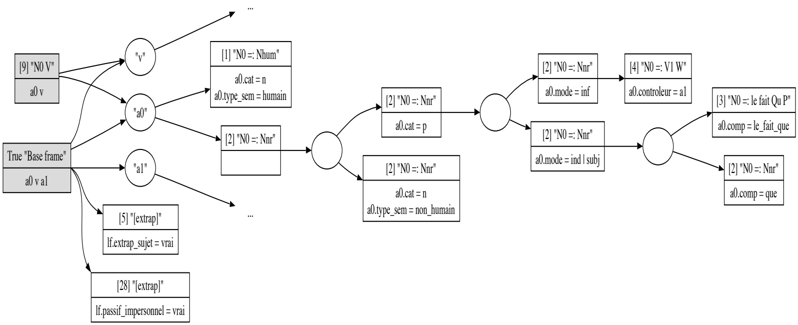
Graph syntax

FRAME-node : grey rectangle with no ingoing edge; 2 parts :

TOP : condition

BOTTOM: indicates the linear order of the argt feature structures compositing the frame

Example graph



Graph processing

1. Given a table graph G, then for each table lign l, a reduced graph is produced.
2. For each reduced graph, the algorithm produces the corresponding lexical entries by enumerating the paths through this graph.

Reduced Graph

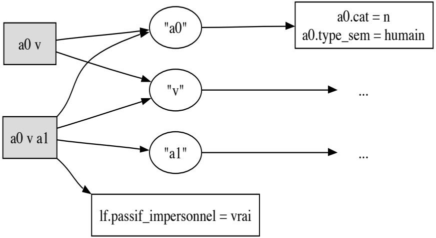
Given a table graph and a lign l in that table, the reduced graph for l is computed as follows:

- for each **AND** and **FRAME** nodes where the condition is `False`, the node and its adjacent edges are suppressed;
- For each **OR** node without outgoing edge, the node and its adjacent nodes are suppressed
- the symbol $\$c$ is replaced by the content of the cell $[l,c]$

In the remaining nodes, the conditions are necessarily `True` and thus can be removed.

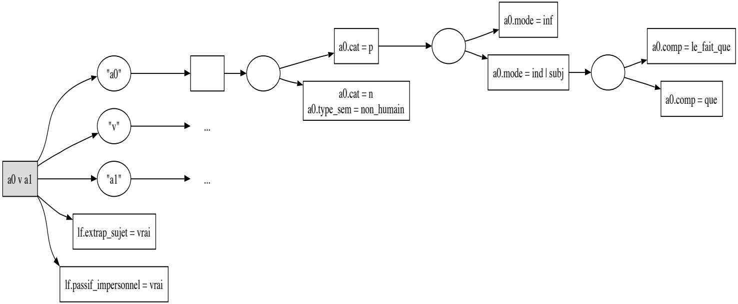
Example (abuser)

	1	2	3	4	5	9	28
	N0=: Nhum	N0=: Nnr	N0=: le fait Qu P	N0=: V1W	[extrap]	N0 V	[extrap] [passif]
abuser	+	−	−	−	−	+	+



Example

	1	2	3	4	5	9	28
	N0=: Nhum	N0=: Nnr	N0=: le fait Qu P	N0=: V1W	[extrap]	N0 V	[extrap] [passif]
bénéficier	−	+	+	−	+	−	+



Graph Processing

From the reduced graph corresponding to each lign of the table, the algorithm computes the lexical entries by computing each possible path as follows:

- for each **FRAME** node F, a frame is initialised of the form $[a_{i1} = \emptyset, a_{i2} = \emptyset, \ldots, a_{ip} = \emptyset, v = \emptyset, a_{j1} = \emptyset, a_{j2} = \emptyset, \ldots, a_{jq} = \emptyset, \mathrm{lf} = \emptyset]$ where $a_{i1}\ a_{i2} \ldots a_{ip}\ v\ a_{j1}\ a_{j2} \ldots a_{jq}$ is given by the bottom part of the frame node F.
- In the subgraph rooted in F, each path is followed and the initial frame F is enriched with the content of the traversed nodes.

Output

```
--abuser--
a0[cat=n, type_sem=humain] v[...]
  lf[passif_impersonnel=vrai]
a0[cat=n, type_sem=humain] v[...] a1[...]
  lf[passif_impersonnel=vrai]

--beneficier--
a0[cat=n, type_sem=non_humain] v[...] a1[...]
  lf[extrap_sujet=vrai, passif_impersonnel=vrai]
a0[cat=p, mode=inf] v[...] a1[...]
  lf[extrap_sujet=vrai, passif_impersonnel=vrai]
a0[cat=p, mode=ind|subj, comp=le_fait_que] v[...] a1[...]
  lf[extrap_sujet=vrai, passif_impersonnel=vrai]
a0[cat=p, mode=ind|subj, comp=que] v[...] a1[...]
  lf[extrap_sujet=vrai, passif_impersonnel=vrai]
```

Quality of the output lexicon

- Table information may be incorrect, superfluous or incomplete
- Conversion process may be incorrect
- No factorisation

Correction facilities

- Modify the graph: a change in the graph is directly reflected in the output lexicon
- Modify the output lexicon
- Modify the graph processing algorithm

We can use these techniques to suppress, add, change and factorise the information contained in the graph.

Deleting information

- Some of the information contained in the LADL tables is usually not present in syntactic lexicons
  - Col. 16, 17: information about compatibility of infinitival complement with temporal adverbials
  - Col. 18, 19, 20: indicate whether infinitival complement can be "must, can" or "know"
- This information can be deleted by filtering the corresponding feature-value pairs out

We use filtering techniques to (i) to derived a simplified lexicon from the LADL tables extracted lexicon and (ii) to derive different formats (in particular, LEFFF compatible format)

Adding information

- The LADL tables do not explicitly specify grammatical functions
- To preserve linking information (i.e., the mapping between grammatical functions and thematic roles), we add this information to the graphs.
- The resulting lexicon contains grammatical function information

Modifying and factorising the information

- Modify the graph or
- Use filtering on the output lexicon

Results and Perspectives

Results: Graphs and lexicons for 12 tables (1 936 verbs and 2 019 entries)
http://www.loria.fr/~gardent/ladl/content/resultats.php

Validation: The output lexicon need to be evaluated.

- Comparison with existing lexicons (LEFFF, syntactic lexicon acquired from corpora)
- Error mining using parser and large corpora (van Noord 2004; de la Clergerie 2006)
- Generation

Extension: create graphs for the other tables

Structuration: factorisation and grouping into Beth Levin type verb classes (aka french VerbNet)