Semantic Interpretation

- **Semantic interpretation**
- **Lexical Semantics**
- **Word Sense Disambiguation (WSD)**

Semantic Interpretation

- Lexical semantics
  - "Pedro" → pedro.
  - “runs" → (lambda(x), run(x)))
- composition
  - (lambda(x), runs(x)) pedro → run(pedro).
- How? Composition functions
- When?

Composition function = lambda evaluation.

SENTENCE → NP VP (2 1)
NP → propernoun, (1)
VP → vi, (1)
VP → vt NP (1 2)

Pedro → propernoun , pedro
María → propernoun, maria
ríe → vi, (lambda (x), run(x))
ama a → vt, ((lambda (x), (lambda (y), loves(y,x))))

Example

"Pedro runs"

```
SENTENCE
|                          SEM = (2 1) = (VP:SEM NP:SEM) =
|                               ((lambda(x) run(x)) pedro) = run(pedro)
|
|________NP
|     |                    SEM = (1) = (propernoun :SEM) = pedro
|     |
|     |________ propernoun
|          |               SEM = pedro
|          |
|               Pedro
|________VP
      |                    SEM = (1) = (vi:SEM) = (lambda(x) run(x))
      |________vi
           |               SEM = (lambda(x) run(x))
           |
          runs
```

Example

"Pedro loves María"

```
SENTENCE
|                         SEM = (2 1) = (VP:SEM NP:SEM) =
|                              ((lambda(x) love(x,maria)) pedro) = love(pedro,maria)
|________NP
|    |                    SEM = (1) = (propernoun :SEM) = pedro
|    |_________ propernoun
|              |               SEM = pedro
|              Pedro
|________VP                    SEM=(1 2) = (vt:SEM NP:SEM) = (((lambda(y)
     |                    (lambda(x) love(x,y))) maria)=((lambda(x)love(x,maria))
     |________vt
     |    |               SEM = ((lambda(y) (lambda(x) love(x,y)))
     |    loves
     |
     |________NP
          |               SEM = (1) = (propernoun :SEM) = maria
          |________ propernoun
              |    SEM = maria
              María
```

Strategies for the Semantic Interpretation

- Once syntactic processing has finished
  - Input: syntactic tree
- Unification of the syntactic and semantic process
  - There is only one process
  - Grammatical categories include semantic information
- Syntactic/semantic processing in parallel
  - Semantic information is used to validate hypothesis of syntactic analysis
  - Partial semantic interpretations and syntactic tree are build at the same time.
- Semantic-guided analyzers
  - Syntactic information (when used) is reduced to validate semantic hypothesis and to categorize part of the text.

Semantic Interpretation Once Syntactic Processing has Finished

- **Allen's proposal**
- Interpretation rules
  - <Schema> → < Partial description of the logical form.>
- The schema consists of a set of restrictions, resulting from the syntactic analysis.
- The final semantic representation is a logical form. From the components of the syntactic tree partial logical forms will be built (including undefined elements).
- Notation:
  - * object (unique identification) described by the logical form
  - t (<descriptor>) Constituent type
  - v (<descriptor>) Semantic interpretation of the constituent
  - ? element not yet assigned.

Syntax of the Logical Form

- (<OPERATOR> <NAME> <TYPE> <MODIFIER>)
  - Operators :
    - Not quantified nominal sintagma
      - DEF/SING, DEF/PL, INDEF/SING, INDEF/PL
    - Quantified nominal sintagma
      - EACH, SOME
    - Interrogative pronoun
      - WH, WH/PL, WH/SING
    - Personal pronoun
      - PRO
    - Proper name
      - NAME
    - Sentence
      - ASSERT, Y/N_QUERY, WH_QUERY, COMMAND
    - Verbal sintagma
      - PAST, PRES, FUT, INF
    - Relation
      - OR, AND, BUT

Examples of interpretation rules

a green apple
(NP DET a
ADJ (green)
HEAD apple)
→
{(? A1 APPLE (COLOR C1 GR137)),
(? A1 APPLE (UNRIPE C1))}

Examples of interpretation rules

```
The man laughed
(S  SUBJ  (NP    DET    the
                 HEAD   man
                 NUM    {3s}
                 SEM    (DEF/SING  M1  MAN))
    MAIN-V              laugh
    TENSE                    past)


→
    (?  L1  LAUGH
            [AGENT  (DEF/SING  M1  MAN)])
```

Algorithm of Semantic Intepretation

- A name is created (it replaces * in the form)
- **The nucleus (sentence head) is analyzed**
  - All possible interpretation rules are applied (those indexed by the nucleus)
  - All possible partial logical forms are combined

- **For each resulting interpretation interpret all modifiers**
  - For each modifier (syntactic component) not yet interpreted
  - All interpretation rules indexed by the modifier head are applied
  - All resulting logical forms are combined using unification

Unification Algorithm 1

- ? can be unified to anything
  - Two identical objects can be unified
  - Two objects including modifiers can be unified if all internal modifiers can be unified.
  - When there are ambigous values unification is done by intersecting the value of patial interpretations.

Unification Algorithm

- Examples
    - (? R1 RUN [THEME (NAME J1 PERSON "John"]))
    - (? R1 RUN [THEME (DEF/SING P1 PRINT-PRESS)])
  - They cannot unify
    - (? R1 RUN [AGENT (NAME J1 PERSON "John"]))
    - (? R1 RUN [THEME (DEF/SING P1 PRINT-PRESS)])
  - If they can, the result is:
    - (? R1 RUN [AGENT (NAME J1 PERSON "John"])) [THEME (DEF/SING P1 PRINT-PRESS)])

Algorithm of semantic interpretation

```
The dishwasher laughed
(S  SUBJ  (NP          DET      the
              NUM      {3s}
              HEAD     diswasher)
          MAIN-V laugh
          TENSE   past)
```

(the.1)	(NP DET the NUM {3s}) → (DEF/SING * ? ?)
(dishwasher.1)	(NP HEAD dishwasher) → (? * {PERSON_TYPE_43, APPLIANCE_TYPE_3} ?)
(laugh.1)	(S SUBJ +animate MAIN_V laugh) → (? * LAUGH [AGENT V(SUBJ)])
(past.1)	(S TENSE PAST) → (PAST ? ? ?)

Algorithm of semantic interpretation

Applying the rule **laugh.1**

==> subj has to include the feature **+animate**

==> **subj** has to be analyzed

==> **np** has to be analyzed

==> application of the **the.1** and **diswasher.1**

```
(DEF/SING  D1  ?  ?)
(?         D1  {PERSON-TYPE-3,
                APPLIANCE-TYPE-43}   ?)
```

Algorithm of semantic interpretation

Partial descriptions are unifyied

The result fills the feature **sem** of the **np** and **subj**.

```
(S   SUBJ    (NP DET      the
                 NUM    {3s}
                 HEAD   diswasher
                 SEM    (DEF/SING   D1
                        {PERSON-TYPE-3,
                         APPLIANCE-TYPE-43})
     MAIN-V             laugh
     TENSE        past)
```

Algorithm of semantic interpretation

The application of the rule **laugh.1** results in

(?L1 LAUGH [AGENT (DEF/SING D1 PERSON-TYPE-3)])

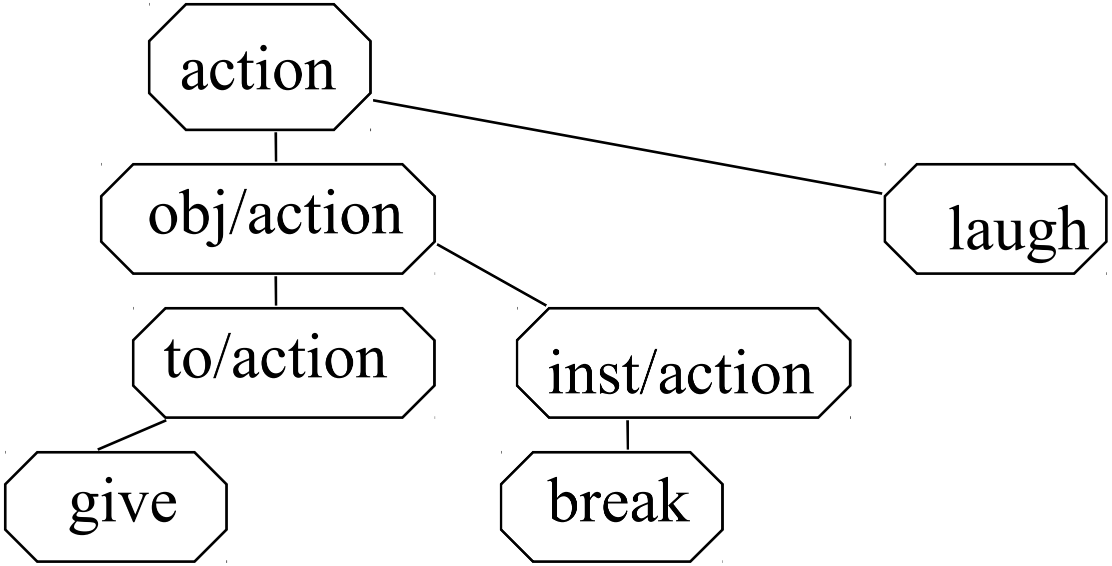
The interpretation of **TENSE**, using the application of the rule **past.1** results in

(PAST ? ? ?)

The final unification results in

(PAST L1 LAUGH [AGENT (DEF/SING D1 PERSON-TYPE-3)])

Rules hierarchy



(give.1)   (S SUBJ +animate MAIN_V give) → (? * GIVE1 [AGENT V(SUBJ)])
(give.2)   (S OBJ   +physobj MAIN_V give) → (? * GIVE1 [THEME V(OBJ)])
(give.3)   (S IOBJ +org MAIN_V give) → (? * GIVE1 [TO_POSS V(IOBJ)])
(give.4)   (S MAIN_V give
MODS (PP PREP to POBJ +org)) → (? * TO_POSS [AGENT V(POBJ)])

Interpretation of the complex components 1

```
The man gave the company a car at the fairgrounds

(S   SUBJ           (NP "the man")
     MAIN-V         give
     TENSE          past
     OBJ      (NP       "a car")
     IOBJ           (NP "the company")
     MODS           ([PP      PREP    at
                    POBJ     (NP "the fairgrounds")]))
```

The application of the algorithm over the nucleus results in:

```
(? G1 GIVE1         [AGENT   (DEF/SING   M1  MAN)]
             [THEME   (INDEF/SING   C1  AUTO)]
             [TO-POS   (DEF/SING   C2   COMPANY_ORG)])
```

Only features TENSE and MODS have not been interpreted

Interpretation of the complex components

The interpretation of the feature *tense* by applying the rule **past.1** results in:

(PAST ? ? ?)

The interpretation of the feature *mods* by applying the rule **at.1** results in:

(? G1 ? (AT-LOC G1 (DEF/SING F1 FAIRGROUND)))

Interpretation of the complex components

The unification of these two forms and the nucleus results in:

```
(PAST   G1   GIVE1 [AGENT   (DEF/SING   M1  MAN)]
     [THEME  (INDEF/SING   C1  AUTO)]
    [TO-POS  (DEF/SING   C2   COMPANY_ORG)]
    (AT-LOC   G1  (DEF/SING  F1 FAIRGROUND)))
```

Semmantic Grammars

- A formalism combining syntax and semantics (usually, context free grammar)
- Terminal symbols correspond to semantic categories
  - The number of rules increases.
  - The number of word for each category decreases.
  - Ambiguity increases.
  - Adaptability decreases

- Application: Robust and efficient systems in restricted domains.
- The logical form representing semantic information is obtained directly from the grammar, using simple extensions.

Semantic Grammars

RES_VP → RESERVING, RES_MODS
DEP_VP → DEPARTING, DEP_MODS

RESERVING → RESERVE_VERB, FLIGHT
RES_MODS → for, PERSON
RES_MODS → []

DEPARTING → DEPART_VERB
DEPARTING → DEPART_VERB, SOURCE_LOC
DEP_MODS → DEP_MOD, DEP_MODS
DEP_MODS → []
DEP_MOD → to, DEST_LOC
DEP_MOD → from, SOURCE_LOC

Lexical Semantics

- Semantic dictionaries
- Ontologies
    - Types
    - Granularity
    - Domain
- Examples
    - UMLS
    - WordNet
    - EuroWordnet
    - Other resources

Example: UMLS

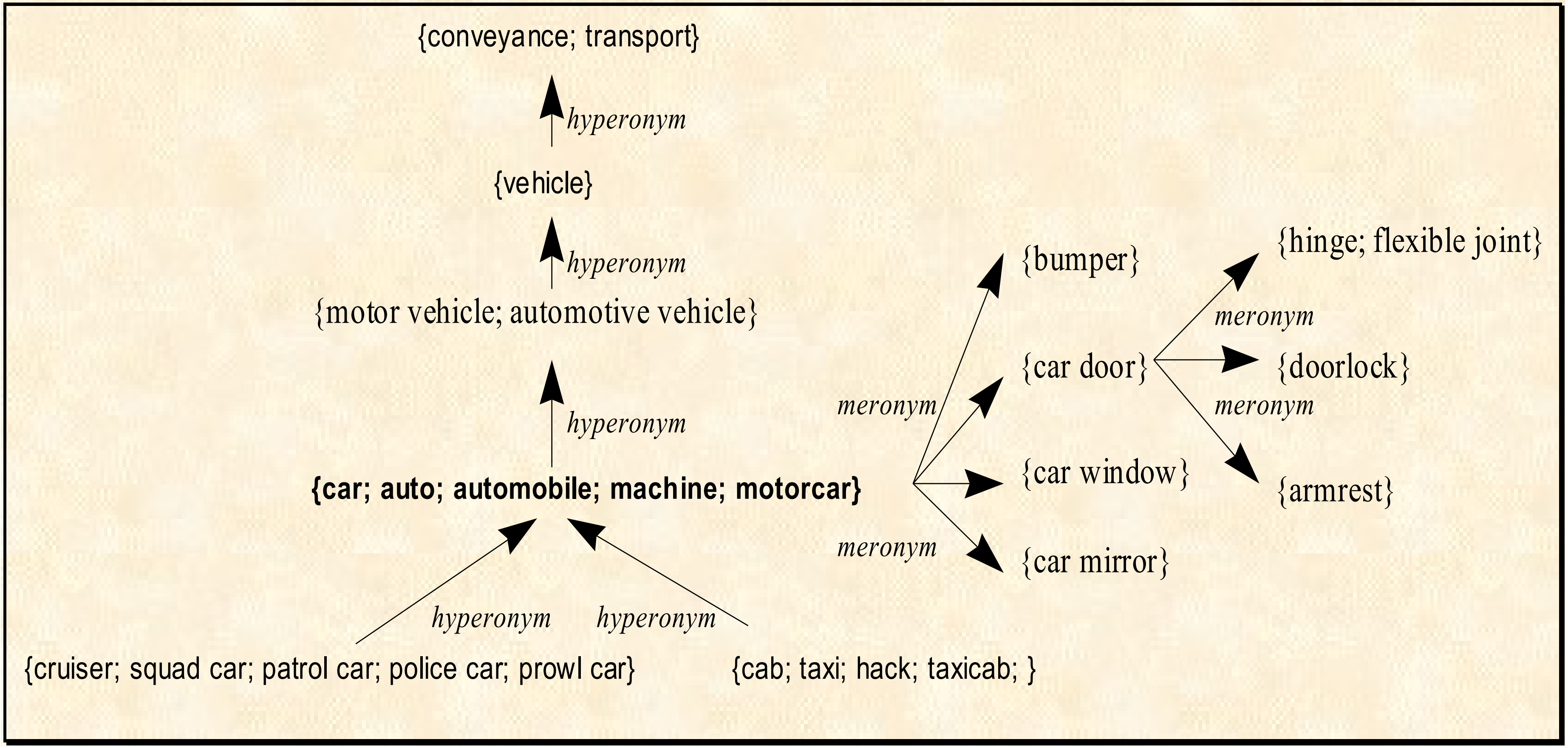
- UMLS (Unified Medical Language System)
  - National Library of Medecine, USA Department of Health and Human Services
  - Resources set
    - Metathesaurus
      - 330.000 concepts, 735.000 terms
    - Semantic Network
      - Set of predefined semantic categories (135 semantic types, 51 relationships)
    - Links to the source vocabularies
      - 30 sources (multilingual)
    - Specialization lexicon with morfo-syntactic information

Example: WordNet

- WordNet
  - University de Princeton (Fellbaum,1998)
  - Lexicalized concepts (words)
    - synsets: set of synonyms
  - Includes nouns, verbs, adjectives and adverbs
  - Related by semantic relationships
    - Sinonima
    - antonyms
    - hyperonym-hyponym
    - implication
    - cause
    - ...
  - Wn1.5, Wn1.6,Wn 1.7: 120.000 wordss, 100.000 synsets
  - Wn2.0, Extended WordNet

http://www.cogsci.princeton.edu/~wn/

Fragment of WN1.5

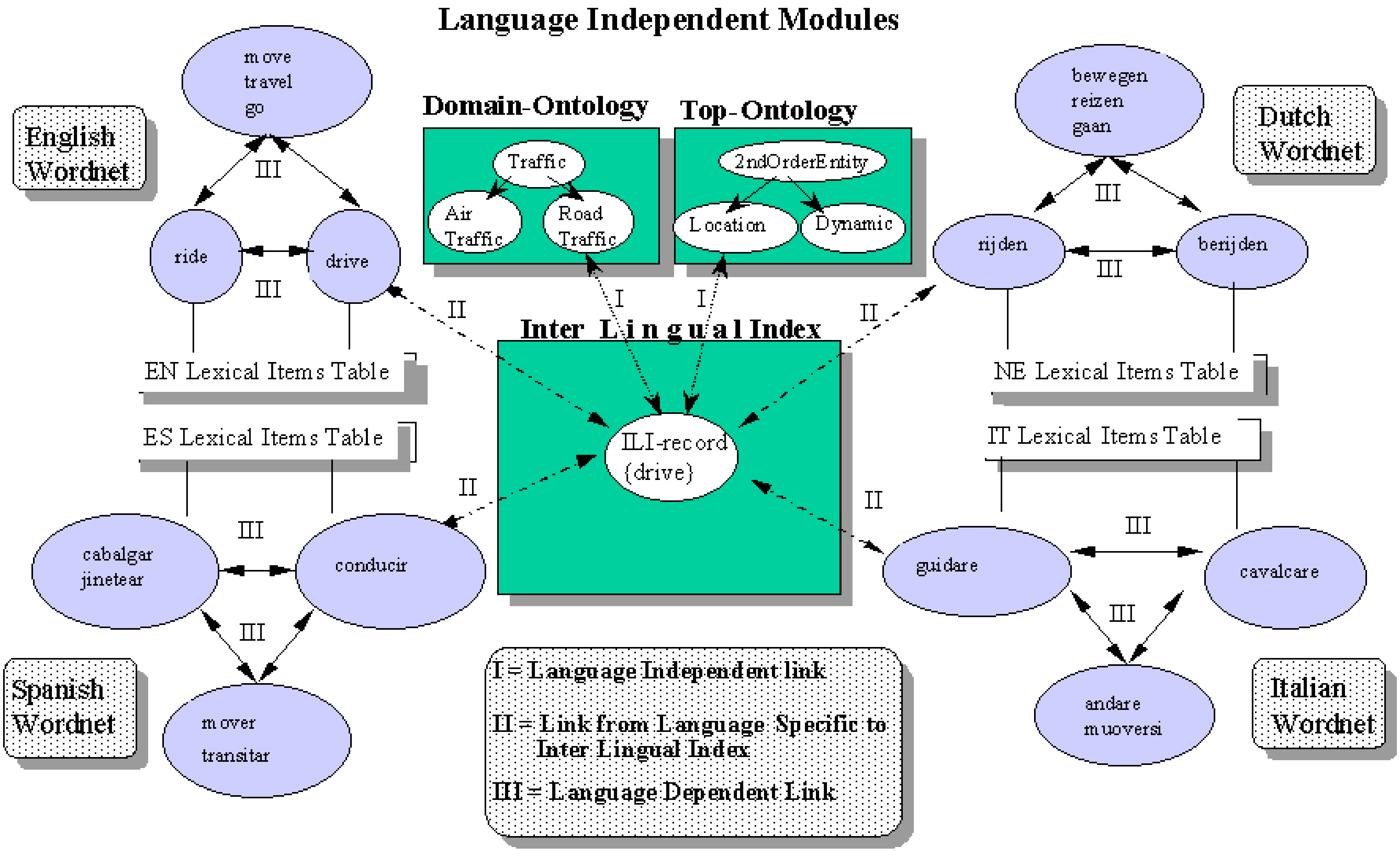


EuroWordNet

- Project LE-2 4003 Telematics Application Programme of the European Community
- Semantic networks in different languages (Integrated)
  - English Universidad de Sheffield
  - Dutch Univ. de Amsterdam
  - Italian I.L.C. de Pisa
  - Spanish UB, UPC, U.N.E.D
- Covers basically nouns and verbs (50.000 meanings for each language)
- Rich in semantic relationships
  - inter/intra lingual, inter/intra category
- EWN2
  - German, Czech, Estonian, French
- Extensions to Catalan, Galician and Basque
- Improvements

http://www.hum.uva.nl/~ewn/
http://www.lsi.upc.es/~nlp/

Architecture of the EuroWordNet Data Structure



Language Independent Modules

Domain-Ontology

Traffic
Air Traffic
Road Traffic

Top-Ontology

2ndOrderEntity
Location
Dynamic

Inter Lingual Index

ILI-record
{drive}

English Wordnet

move
travel
go

ride
drive

EN Lexical Items Table

Dutch Wordnet

bewegen
reizen
gaan

rijden
berijden

NE Lexical Items Table

Spanish Wordnet

ES Lexical Items Table

cabalgar
jinetear

conducir

mover
transitar

Italian Wordnet

IT Lexical Items Table

guidare
cavalcare

andare
muoversi

I = Language Independent link

II = Link from Language Specific to Inter Lingual Index

III = Language Dependent Link

Levin classes (3100 verbs)

- 47 top level classes, 193 second and third level

- Based on pairs of sintactic patterns.

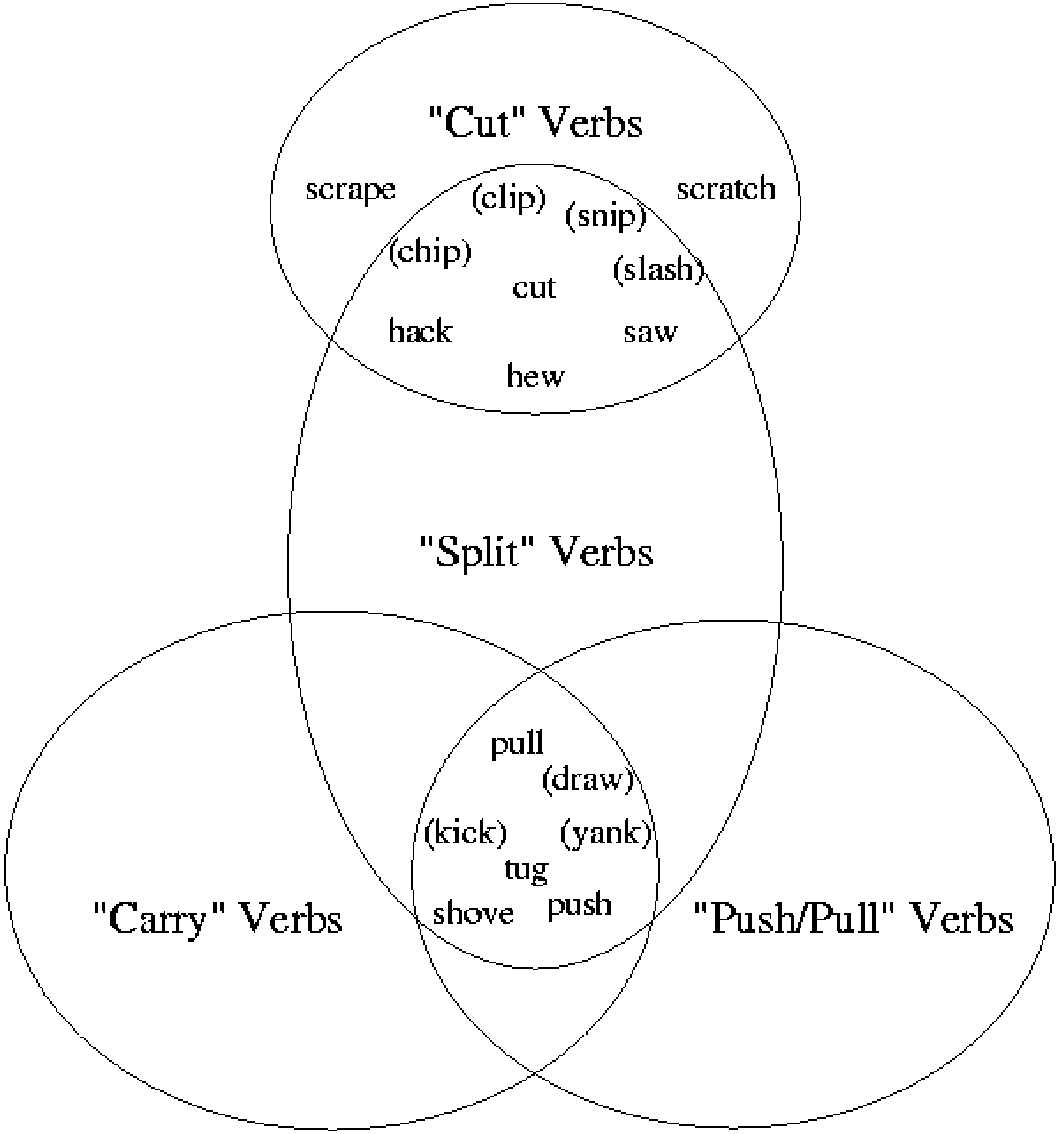
  John broke the jar. / Jars break easily. / The jar broke.
  *John cut the bread. / Bread cuts easily. / *The bread cut.*
  *John hit the wall. / *Walls hit easily. / *The wall hit.*

- Reflect implicit semantic component

  contact, directed motion,
  exertion of force, change of state

- Synonymy, syntactics patterns (subcategorization patterns)

Intersective Levin classes



Regular Sense Extensions

John pushed the chair. **+force, +contact**

John pushed the chairs apart. **+ch-state**

John pushed the chairs across the room. **+ch-loc**

John pushed at the chair. **-ch-loc**

The train whistled into the station. **+ch-loc**

The truck roared past the weigh station. **+ch-loc**

VerbNet 1

- Computational Verbal lexicon
- Associations between syntax and semantics
  - Syntactic frames (subcategorization patterns) and selection restrictions.
  - Lexical semantic information– predicate/argument structure
  - Semantic components represented as predicates and Links to WordNet synsets
- Entries based on precise description of Levin classes
- Temporales properties represented in an explicity form
  - during(E), end(E), result(E)

VerbNet 2

Hit Class

<<MEMBERS>>	[<hit1>, <kick1>,<slap1>…]
<<THEMATIC ROLES>>	Agent(A), Patient(P), Instrument(I)
<<SELECT RESTRICTIONS>>	Agent [+animate], Patient [+concrete], Instrument [+concrete,-animate]

<<FRAMES and PREDICATES>>

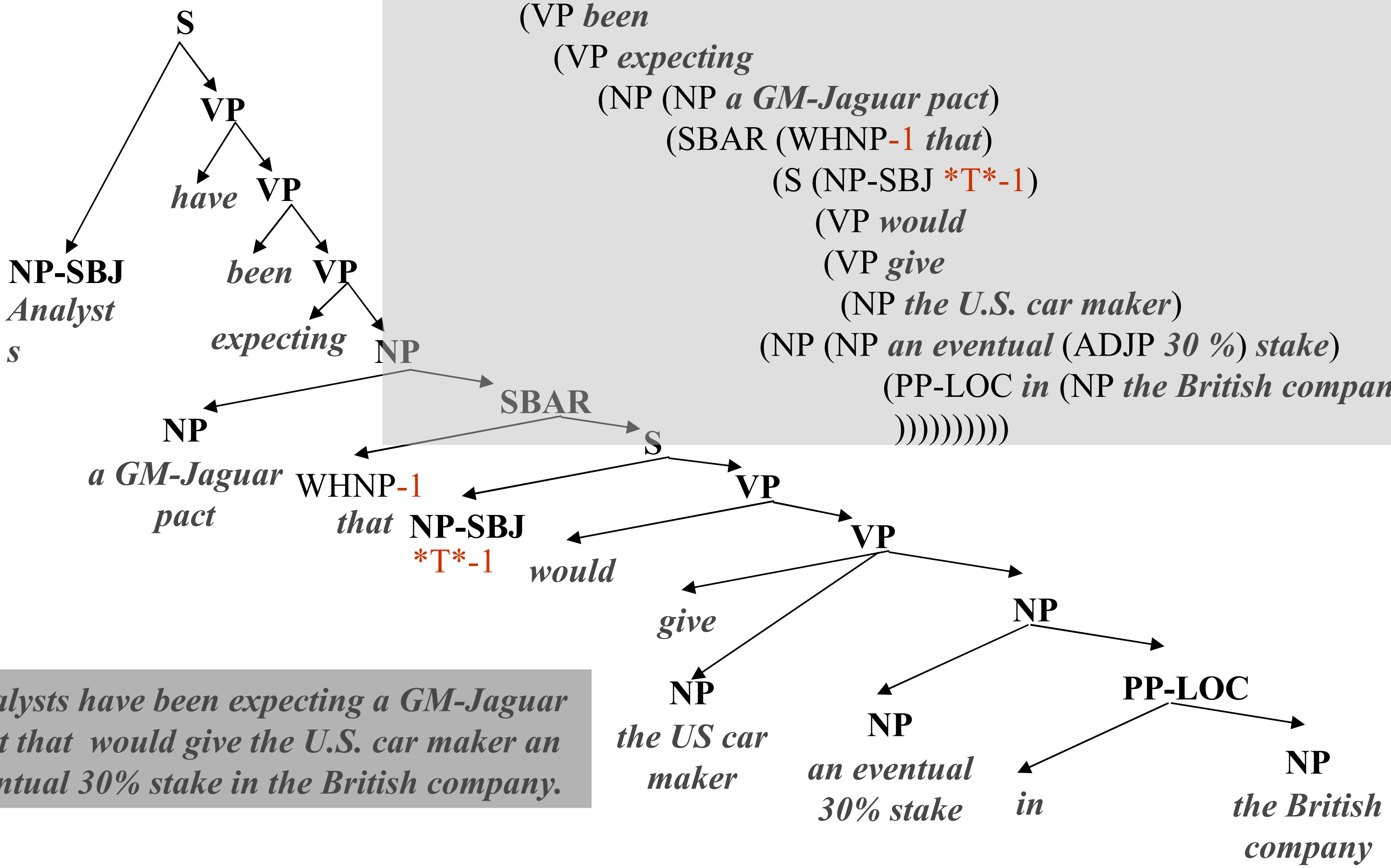
Basic Transitive	A V P	Manner (during(E), directedmotion,A)^ Manner (end(E), forceful,A)^ Contact(end(E),A,P)
Transitive with Instrument	AVP with I	Manner (during (E), directedmotion,I)^ Manner (end (E),forceful,I)^ Contact (end(E),I,P)
Conative	AV at P	Manner (during (E), directedmotion, A)
With/against alternation	A V I against/on P	Manner(during (E), directedmotion, I)^ Manner(end(E), forceful, I)^ Contact (end(E), I, P)

Penn Treebank

- 1,3 millon wods, 40.000 sentences
- Wall Street Journal and other sources
- POS tagged
- Syntactically Parsed

A TreeBanked Sentence

```
(S (NP-SBJ Analysts)
    (VP have
       (VP been
          (VP expecting
              (NP (NP a GM-Jaguar pact)
                    (SBAR (WHNP-1 that)
                              (S (NP-SBJ *T*-1)
                                  (VP would
                                   (VP give
                                     (NP the U.S. car maker)
                           (NP (NP an eventual (ADJP 30 %) stake)
                                     (PP-LOC in (NP the British company))
                                      ))))))))))
```



S
VP
have **VP**
NP-SBJ
Analysts
been **VP**
expecting **NP**
NP
a GM-Jaguar pact
SBAR
WHNP-1
that
S
NP-SBJ
T-1
VP
would
VP
give
NP
the US car maker
NP
NP
an eventual 30% stake
PP-LOC
in
NP
the British company

Analysts have been expecting a GM-Jaguar pact that would give the U.S. car maker an eventual 30% stake in the British company.

Proposition Bank (Propbank)

Generalization of sentences tp propositions

Powell met Zhu Rongji

Powell and Zhu Rongji met

Powell met with Zhu Rongji

Powell and Zhu Rongji had a meeting

. . .

battle

wrestle

join

debate

consult

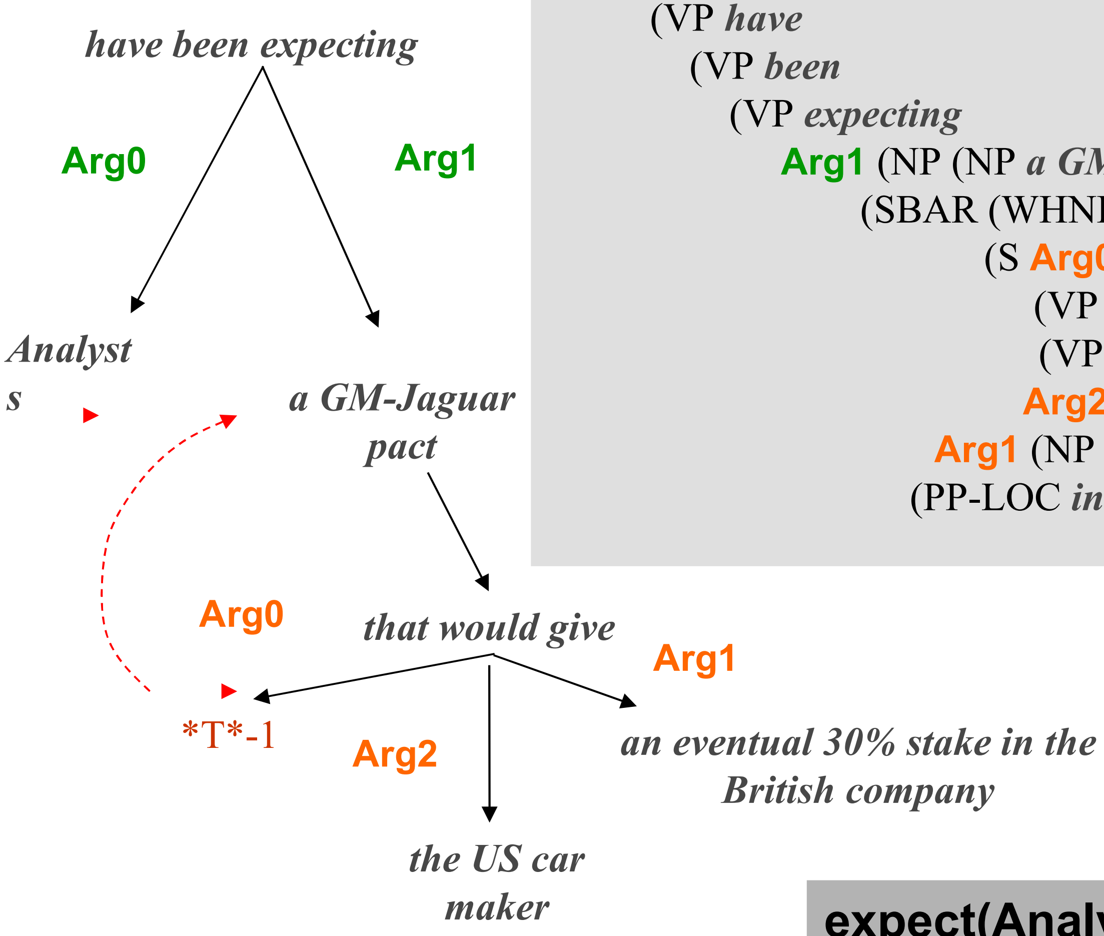
Proposition: **meet**(Powell, Zhu Rongji)

meet(Somebody1, Somebody2)

When Powell met Zhu Rongji on Thursday they discussed the return of the spy plane.

meet(Powell, Zhu)   discuss([Powell, Zhu], return(X, plane))

The same sentence, PropBanked


have been expecting
Arg0
Arg1
Analysts
a GM-Jaguar pact
Arg0
that would give
Arg1
T-1
Arg2
an eventual 30% stake in the British company
the US car maker

```
(S Arg0 (NP-SBJ Analysts)
   (VP have
      (VP been
         (VP expecting
            Arg1 (NP (NP a GM-Jaguar pact)
                  (SBAR (WHNP-1 that)
                        (S Arg0 (NP-SBJ *T*-1)
                           (VP would
                            (VP give
                             Arg2 (NP the U.S. car maker)
                        Arg1 (NP (NP an eventual (ADJP 30 %) stake)
                  (PP-LOC in (NP the British company))
                                 ))))))))
```

expect(Analysts, GM-J pact)
give(GM-J pact, US car maker, 30% stake)

Sense Desambiguation 1

- Senses
  - Different meanings of one word (word type) observable in different uses (word tokens)
- Word Sense Disambiguation (WSD)
  - Given a word (word token) in a context determine its correct sense

Sense Desambiguation 2

- Problems
  - Homonymy: meanings not related
  - Polisemy: related meanings
  - Metaphor
  - Is it possible?

Sense Desambiguation 3

- A similar problem to POS tagging desambiguation
  - Similar approaches
  - A more complex problem
- Common restrictions
  - Yarowsky (1995)
    - One sense per discourse
    - One sense per collocation

Sense Desambiguation 4

- A possible solution (Naive Bayes)
  - Being **w** the word to desambiguate
  - Being c_k the possible senses
  - Being $\vec{x}$ the context vector (i.e. A fixed windows of 100 words).

$$P(c_k|\vec{x})$$

  - When applying Bayes results:

Sense Desambiguation 5

$$c' = \arg\max_{c_k} P(c_k \mid \vec{x}) =$$

$$\arg\max_{c_k} \frac{P(\vec{x} \mid c_k)}{P(\vec{x})} \cdot P(c_k) =$$

$$\arg\max_{c_k} P(x \mid c_k) \cdot P(c_k) =$$

$$\arg\max_{c_k} \left[\log P(\vec{x} \mid c_k) + \log P(c_k)\right] =$$

$$\arg\max_{c_k} \left[\sum_{v_j \ in \ \vec{x}} \log P(v_j \mid c_k) + \log P(c_k) \right]$$