Language Processing with Perl and Prolog
Chapter 15: Lexical Semantics

Pierre Nugues

Lund University
Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/
Words and Meaning

Referred to as lexical semantics:

- Classes of words: If it is hot, can it be cold?
- Definition What is a meal? What is table?
- Reasoning: The meal is on the table. Is it cold?
Categories of Words

Expressions, which are in no way composite, signify substance, quantity, quality, relation, place, time, position, state, action, or affection. To sketch my meaning roughly, examples of substance are ‘man’ or ‘the horse’, of quantity, such terms as ‘two cubits long’ or ‘three cubits long’, of quality, such attributes as ‘white’, ‘grammatical’. ‘Double’, ‘half’, ‘greater’, fall under the category of relation; ‘in the market place’, ‘in the Lyceum’, under that of place; ‘yesterday’, ‘last year’, under that of time. ‘Lying’, ‘sitting’, are terms indicating position, ‘shod’, ‘armed’, state; ‘to lance’, ‘to cauterize’, action; ‘to be lanced’, ‘to be cauterized’, affection.

Aristotle, Categories, IV. (trans. E. M. Edghill)
Representation of Categories

expressions

substance  quantity  quality  relation  place  time  position  state  action  affection
Classes

- Synonymy/Antonymy
- Polysemy
- Hyponyms/Hypernyms is_a(tree, plant), life form, entity
- Meronyms/Holonyms part_of(leg, table)
- Grammatical cases: \[\text{nominative I broke}\] \[\text{accusative the window}\] \[\text{ablative with a hammer}\]
- Semantic cases: \[\text{actor I broke}\] \[\text{object the window}\] \[\text{instrument with a hammer}\]
- Case ambiguity (The window broke/ I broke the window)
%% is_a(Word, Hypernym)
\begin{verbatim}
is_a(hedgehog, insectivore).
is_a(cat, feline).
is_a(feline, carnivore).
is_a(insectivore, mammal).
is_a(carnivore, mammal).
is_a(mammal, animal).
is_a(animal, animate_being).
\end{verbatim}

hyernym(X, Y) :- is_a(X, Y).
hyernym(X, Y) :- is_a(X, Z), hyernym(Z, Y).
Semantic Networks

- substance
  - animates
    - human beings
    - animals
      - eat
      - possess
      - meat
        - eat
        - mammals
          - insectivores
          - carnivores
          - food
        - furniture
An Example: WordNet

Nouns
- hyponyms/hypernyms
- synonyms/antonyms
- meronyms

Adjectives
- synonyms/antonyms
- relational fraternal $\rightarrow$ brother

Verbs
- Semantic domains (body function, change, communication, perception, contact, motion, creation, possession, competition, emotion, cognition, social interaction, weather)
- Synonymy, Antonymy: (rise/fall, ascent/descent, live/die)
- “Entailment”: succeed/try, snore/sleep
The caterpillar ate the hedgehog.

Representation:

\[ \exists (X, Y), \text{caterpillar}(X) \land \text{hedgehog}(Y) \land \text{ate}(X, Y). \]

Reasoning (inference):
It is untrue because the query:

?- predator(X, hedgehog)
X = foxes, eagles, car drivers, ...

but no caterpillar.
Lexicons

Words are ambiguous: A same form may have more than one entry and sense. The *Oxford Advanced Learner’s Dictionary* (OLAD) lists five entries for *bank*:

1. *noun*, raised ground
2. *verb*, turn
3. *noun*, organization
4. *verb*, place money
5. *noun*, row or series

and five senses for the first entry.
Definitions

Short texts describing a word:

- A genus or superclass using a hypernym.
- Specific attributes to differentiate it from other members of the superclass. This part of the definition is called the differentia specifica.

bank (1.1): a land sloping up along each side of a canal or a river.
hedgehog: a small animal with stiff spines covering its back.
waiter: a person employed to serve customers at their table in a restaurant, etc.
## Significance of the Sense

<table>
<thead>
<tr>
<th>French</th>
<th>German</th>
<th>Danish</th>
</tr>
</thead>
<tbody>
<tr>
<td>arbre</td>
<td>Baum</td>
<td>Træ</td>
</tr>
<tr>
<td>bois</td>
<td>Holz</td>
<td></td>
</tr>
<tr>
<td>forêt</td>
<td>Wald</td>
<td>Skov</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>French</th>
<th>Welsh</th>
</tr>
</thead>
<tbody>
<tr>
<td>vert</td>
<td>gwyrrdd</td>
</tr>
<tr>
<td>bleu</td>
<td>glas</td>
</tr>
<tr>
<td>gris</td>
<td>llwyd</td>
</tr>
<tr>
<td>brun</td>
<td></td>
</tr>
</tbody>
</table>
Sense Tagging Using the Oxford Advanced Learner’s Dictionary (OALD)

Sentence: *The patron ordered a meal*

<table>
<thead>
<tr>
<th>Words</th>
<th>Definitions</th>
<th>Sense</th>
</tr>
</thead>
<tbody>
<tr>
<td>The patron</td>
<td><strong>Correct sense:</strong> A customer of a shop, restaurant, theater</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td><strong>Alternate sense:</strong> A person who gives money or support to a person, an organization, a cause or an activity</td>
<td>1.1</td>
</tr>
<tr>
<td>ordered</td>
<td><strong>Correct sense:</strong> To request somebody to bring food, drink, etc in a hotel, restaurant etc.</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td><strong>Alternate senses:</strong> To give an order to somebody</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>To request somebody to supply or make goods, etc.</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>To put something in order</td>
<td></td>
</tr>
<tr>
<td>a meal</td>
<td><strong>Correct sense:</strong> The food eaten on such occasion</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Alternate sense:</strong> An occasion where food is eaten</td>
<td>1.1</td>
</tr>
</tbody>
</table>
Identifying Senses

Semantic tagging looks like POS tagging: it assumes the sense of a word depends on its context.

We analyze the interaction between bank and market finance in a model where bankers gather information through monitoring...

Statistical techniques optimize a sequence of semantic tags. The context $C$ of word $w$ is defined as:

$$w_{-m}, w_{-m+1}, \ldots, w_{-1}, w, w_1, \ldots, w_{m-1}, w_m.$$ 

If $w$ has $n$ senses, $s_1 \ldots s_n$, the optimal sense given $C$ is defined as:

$$\hat{s} = \arg\max_{s_i, 1 \leq i \leq n} P(s_i | C).$$

Using Bayes’ rule, we have:

$$\hat{s} = \arg\max_{s_i, 1 \leq i \leq n} P(s_i) P(C | s_i),$$

$$= \arg\max_{s_i, 1 \leq i \leq n} P(s_i) P(w_{-m}, w_{-m+1}, \ldots, w_{-1}, w_1, \ldots, w_{m-1}, w_m | s_i).$$
The Naïve Bayes classifier uses the bag-of-word approach. We replace
\[ P(w_{-m}, w_{-m+1}, \ldots, w_{-1}, w_1, \ldots, w_{m-1}, w_m | s_i) \]
with the product of probabilities:
\[ \prod_{j=-m, j \neq 0}^{m} P(w_j | s_i). \]

SemCor is a sense-annotated corpus for English. Semisupervised and unsupervised algorithms
We analyze the interaction between bank and market finance in a model where bankers gather information through monitoring and screening.

Maximally overlapping definitions (Oxford Advanced Learner’s Dictionary, 1995):

- **Bank:**
  
  Sense 1: The land sloping up along each side of a river or a canal; the ground near a river
  
  Sense 3: An organization or a place that provides a financial service. Customers keep their money in the bank safely and it is paid out when needed by the means of cheques, etc.

- **Finance:**
  
  Sense 1: The money used or needed to support an activity, project, etc; the management of money
Dictionaries store information about how words combine with other words to form larger structures. This information is called valence (cf. valence in chemistry). In the *Oxford Advanced Learner’s Dictionary*, *tell*, sense 1, has the valence patterns:
tell something (to somebody) / tell somebody (something)
as in:
- *I told a lie to him*
- *I told him a lie*

Both have the same predicate–argument representation:
tell.01(Speaker: I, Utterance: a lie, Hearer: him)
Case Grammar

Verbs have semantic cases (or semantic roles):
- An Agent – Instigator of the action (typically animate)
- An Instrument – Cause of the event or object in causing the event (typically animate)
- A Dative – Entity affected by the action (typically animate)
- A Factitive – Object or being resulting from the event
- A Locative – Place of the event
- A Source – Place from which something moves,
- A Goal – Place to which something moves,
- A Beneficiary – Being on whose behalf the event occurred (typically animate)
- A Time – Time at which the event occurred
- An Object – Entity that is acted upon or that changes, the most general case.
Case Grammar: An Example

open(Object, \{Agent\}, \{Instrument\})

The door opened
John opened the door
The wind opened the door
John opened the door with a chisel

Object = door
Object = door and Agent = John
Object = door and Agent = wind
Object = door, Agent = John, and Instrument = chisel
The waiter brought the meal to the patron

Identify the verb **bring** and apply constraints:

<table>
<thead>
<tr>
<th>Case</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agentive</td>
<td>Animate</td>
<td>The waiter</td>
</tr>
<tr>
<td>Objective (or theme)</td>
<td>Animate</td>
<td>the meal</td>
</tr>
<tr>
<td>Dative</td>
<td>Animate</td>
<td>the patron</td>
</tr>
<tr>
<td>Time</td>
<td>Animate</td>
<td>past</td>
</tr>
</tbody>
</table>
In 1968, Fillmore wrote an oft cited paper on case grammars. Later, he started the FrameNet project: http://framenet.icsi.berkeley.edu/
Framenet is an extensive lexical database itemizing the case (or frame) properties of English verbs. In FrameNet, Fillmore no longer uses universal cases but a set of frames – predicate argument structures – where each frame is specific to a class of words.
The Impact Frame

Impact:

*bang.v, bump.v, clang.v, clunk.v, collide.v, collision.n, crash.v, crash.n, crunch.v, glancing.a, graze.v, hit.v, hit.n, impact.v, impact.n, plop.v, plough.v, plunk.v, run.v, slam.v, slap.v, smack.v, smash.v, strike.v, thud.v, thump.v*

Frame elements:

*cause, force, impactee, impactor, impactors, manner, place, result, speed, sub_location, time.*
The Revenge Frame

15 lexical units (verb, nouns, adjectives):

avenge.v, avenger.n, get back (at).v, get_even.v, retaliate.v, retaliation.n, retribution.n, retributive.a, retributory.a, revenge.n, revenge.v, revengeful.a, revenger.n, vengeance.n, vengeful.a, and vindictive.a.

Five frame elements (FE):

Avenger, Punishment, Offender, Injury, and Injured_party.

The lexical unit in a sentence is called the target.
Annotation

2. With this, [Avenger] El Cid at once avenged [Injury] the death of his son.

FrameNet uses three annotation levels: Frame elements, Phrase types (categories), and grammatical functions. GFs are specific to the target’s part-of-speech (i.e. verbs, adjectives, prepositions, and nouns).

For the verbs, three GFs: Subject (Ext), Object (Obj), Complement (Dep), and Modifier (Mod), i.e. modifying adverbs ended by –ly or indicating manner.
**The Valence Pattern**

<table>
<thead>
<tr>
<th>Sent. 1</th>
<th><em>avenge</em></th>
<th>FE</th>
<th>Avenger</th>
<th>Injured_party</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PT</td>
<td>NP</td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GF</td>
<td>Ext</td>
<td>Object</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sent. 2</th>
<th><em>avenge</em></th>
<th>FE</th>
<th>Avenger</th>
<th>Injury</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PT</td>
<td>NP</td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GF</td>
<td>Ext</td>
<td>Obj</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sent. 3</th>
<th><em>avenge</em></th>
<th>FE</th>
<th>Avenger</th>
<th>Injured_party</th>
<th>Offender</th>
<th>Punishment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PT</td>
<td>NP</td>
<td>NP</td>
<td>PP</td>
<td>PPing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GF</td>
<td>Ext</td>
<td>Obj</td>
<td>Comp</td>
<td>Comp</td>
</tr>
</tbody>
</table>
Given a sentence:

\[ I \text{ told him a lie} \]

and a target word – `tell` –, find the semantic arguments.

In Propbank, the possible arguments of `tell.01` are *speaker* (Arg0), *utterance* (Arg1), and *hearer* (Arg2).

Input: a syntax tree:
Two steps:

- Find the arguments,
- Determine the role (name) of each argument

The identification of semantic arguments can be modeled as a statistical classification problem.

What features are useful for this task? Examples:

- Grammatical function: subject, object, ...
- Voice: I told a lie / I was told a lie
- Semantic classes: I told him / the note told him
- Semantic class usually not available: use word instead
Given a dependency tree:

```
<root>
  \_ I
    \_ told
      \_ him
        \_ a
          \_ lie
            \_ noun
```

We select the three dependents of *told* and we extract features to determine if it is a semantic argument and its name.

<table>
<thead>
<tr>
<th>Word</th>
<th>Grammatical function</th>
<th>Voice</th>
<th>Argument</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>I</em></td>
<td>Subject</td>
<td>Active</td>
<td><em>speaker</em> (Arg0)</td>
</tr>
<tr>
<td><em>him</em></td>
<td>Indirect object</td>
<td>Active</td>
<td><em>hearer</em> (Arg2)</td>
</tr>
<tr>
<td><em>lie</em></td>
<td>Direct object</td>
<td>Active</td>
<td><em>utterance</em> (Arg1)</td>
</tr>
</tbody>
</table>
Semantic analysis often uses Propbank instead of Framenet because of Propbank’s larger annotated corpus. CoNLL 2008 and 2009 used Propbank for their evaluation of semantic parsers.

CoNLL annotation format of the sentence:

_The luxury auto maker last year sold 1,214 cars in the U.S._

<table>
<thead>
<tr>
<th>ID</th>
<th>Form</th>
<th>Lemma</th>
<th>PLemma</th>
<th>POS</th>
<th>PPOS</th>
<th>Feats</th>
<th>PFeats</th>
<th>Head</th>
<th>PHead</th>
<th>Deprel</th>
<th>PDeprel</th>
<th>FillPred</th>
<th>Sense</th>
<th>APred1</th>
<th>APred2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The</td>
<td>the</td>
<td>the</td>
<td>DT</td>
<td>DT</td>
<td>_</td>
<td>_</td>
<td>4</td>
<td>4</td>
<td>NMOD</td>
<td>NMOD</td>
<td>_</td>
<td>_</td>
<td>A1</td>
<td>_</td>
</tr>
<tr>
<td>2</td>
<td>luxury</td>
<td>luxury</td>
<td>luxury</td>
<td>NN</td>
<td>NN</td>
<td>_</td>
<td>_</td>
<td>3</td>
<td>3</td>
<td>NMOD</td>
<td>NMOD</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>3</td>
<td>auto</td>
<td>auto</td>
<td>auto</td>
<td>NN</td>
<td>NN</td>
<td>_</td>
<td>_</td>
<td>4</td>
<td>4</td>
<td>NMOD</td>
<td>NMOD</td>
<td>_</td>
<td>_</td>
<td>A1</td>
<td>_</td>
</tr>
<tr>
<td>4</td>
<td>maker</td>
<td>maker</td>
<td>maker</td>
<td>NN</td>
<td>NN</td>
<td>_</td>
<td>_</td>
<td>7</td>
<td>7</td>
<td>SBJ</td>
<td>SBJ</td>
<td>_</td>
<td>Y</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>5</td>
<td>last</td>
<td>last</td>
<td>last</td>
<td>JJ</td>
<td>JJ</td>
<td>_</td>
<td>_</td>
<td>6</td>
<td>6</td>
<td>NMOD</td>
<td>NMOD</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>6</td>
<td>year</td>
<td>year</td>
<td>year</td>
<td>NN</td>
<td>NN</td>
<td>_</td>
<td>_</td>
<td>7</td>
<td>7</td>
<td>TMP</td>
<td>TMP</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>7</td>
<td>sold</td>
<td>sell</td>
<td>sell</td>
<td>VBD</td>
<td>VBD</td>
<td>_</td>
<td>_</td>
<td>0</td>
<td>0</td>
<td>ROOT</td>
<td>ROOT</td>
<td>_</td>
<td>Y</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>8</td>
<td>1,214</td>
<td>1,214</td>
<td>1,214</td>
<td>CD</td>
<td>CD</td>
<td>_</td>
<td>_</td>
<td>9</td>
<td>9</td>
<td>NMOD</td>
<td>NMOD</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>A1</td>
</tr>
<tr>
<td>9</td>
<td>cars</td>
<td>car</td>
<td>car</td>
<td>NNS</td>
<td>NNS</td>
<td>_</td>
<td>_</td>
<td>7</td>
<td>7</td>
<td>OBJ</td>
<td>OBJ</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>AM-LOC</td>
</tr>
<tr>
<td>10</td>
<td>in</td>
<td>in</td>
<td>in</td>
<td>IN</td>
<td>IN</td>
<td>_</td>
<td>_</td>
<td>7</td>
<td>7</td>
<td>LOC</td>
<td>LOC</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>11</td>
<td>the</td>
<td>the</td>
<td>the</td>
<td>DT</td>
<td>DT</td>
<td>_</td>
<td>_</td>
<td>12</td>
<td>12</td>
<td>NMOD</td>
<td>NMOD</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>12</td>
<td>U.S.</td>
<td>u.s.</td>
<td>u.s.</td>
<td>NNP</td>
<td>NNP</td>
<td>_</td>
<td>_</td>
<td>10</td>
<td>10</td>
<td>PMOD</td>
<td>PMOD</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
</tbody>
</table>
Visualizing Dependencies

Syntactic dependencies:

Semantic dependencies (predicate–argument structures):

- Root-
- The
- luxury
- auto
- maker
- last
- year
- sold
- 1,214
- cars
- in
- the
- U.S.
The luxury auto maker last year sold 1,214 cars in the U.S.
The luxury auto maker last year sold 1,214 cars in the U.S.

**Predicate identification**

The luxury auto *maker* last year *sold* 1,214 cars in the U.S.  
  - maker: `maker.01`  
  - sell: `sell.01`

**Predicate sense disambiguation**

The luxury auto *maker* last year *sold* 1,214 cars in the U.S.  
  - maker: `maker.01`  
  - sell: `sell.01`

**Argument identification**

The luxury auto *maker* last year *sold* 1,214 cars in the U.S.  
  - sell: `sell.01`

**Argument labeling**

The luxury auto *maker* last year *sold* 1,214 cars in the U.S.  
  - A0
  - AM-TMP
  - sell: `sell.01`
  - A1
  - AM-LOC
Almost all the semantic parsers (or semantic role labelers) start with a parsing step: either dependencies or constituents. The semantic parser consists of a sequence of classifiers. Logistic regression is among the best classifiers. Each classifier uses a set of features extracted from the previous steps.
# Features for the Predicate Identification

Features used by Johansson and Nugues (2008) and values for *sold* in *The luxury auto maker last year sold 1,214 cars in the U.S.*

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PredForm</td>
<td>sold</td>
</tr>
<tr>
<td>PredLemma</td>
<td>sell</td>
</tr>
<tr>
<td>PredHeadForm</td>
<td>ROOT</td>
</tr>
<tr>
<td>PredHeadPOS</td>
<td>ROOT</td>
</tr>
<tr>
<td>PredDeprel</td>
<td>ROOT</td>
</tr>
<tr>
<td>ChildFormSet</td>
<td>{maker, year, cars, in}</td>
</tr>
<tr>
<td>ChildPOSSet</td>
<td>{NN, NNS, IN}</td>
</tr>
<tr>
<td>ChildDepSet</td>
<td>{SBJ, TMP, OBJ, LOC}</td>
</tr>
<tr>
<td>DepSubcat</td>
<td>SBJ+TMP+OBJ+LOC</td>
</tr>
<tr>
<td>ChildFormDepSet</td>
<td>{maker+SBJ, year+TMP, cars+OBJ, in+LOC}</td>
</tr>
<tr>
<td>ChildPOSDepSet</td>
<td>{NN+SBJ, NN+TMP, NNS+OBJ, IN+LOC}</td>
</tr>
</tbody>
</table>
EVAR is a German project that aims at providing information on trains.
EVAR’s Case Grammar

1. **fahren1.1** (*The train is going from Hamburg to Munich*)
   - Instrument: noun group (nominative), Transport, obligatory
   - Source: prepositional group (Origin), Location, optional
   - Goal: prepositional group (Direction), Location, optional

2. **fahren1.2** (*I am going by train from Hamburg to Munich*)
   - Agent: noun group (nominative), Animate, obligatory
   - Instrument: prepositional group (prep = mit), Transport, optional
   - Source: prepositional group (Origin), Location, optional
   - Goal: prepositional group (Direction), Location, optional

3. **Abfahrt1.1** (*The departure of the train at Hamburg for Munich*)
   - Object: noun group (genitive), Transport, optional
   - Location: prepositional group (Place), Location, optional
   - Time: prepositional group (Moment), Time, optional
Application: Carsim

Identify the events (actions) and the semantic relations related to car accidents.

In Framenet, the **Impact** class consists of 38 verbs or nouns with the roles: **Impactor**, **Impactee**, **Impactees**

\[
\text{\langle Impactor\rangle The rock \text{ HIT} \langle Impactee\rangle the sand} \text{ with a thump}
\]

Source: [http://framenet.icsi.berkeley.edu/](http://framenet.icsi.berkeley.edu/)

In Carsim:

\[
\text{\langle ACTOR\rangle En personbil \text{ körde} \langle TIME\rangle vid femtiden | \langle TIME\rangle på torsdagseftermiddagen} \text{ in \langle VICTIM\rangle i ett radhus | \langle LOC\rangle i ett äldreboende | \langle LOC\rangle på Alvägen | \langle LOC\rangle i Enebyberg | \langle LOC\rangle norr om Stockholm}.
\]