

Language Technology

<http://cs.lth.se/edan20/>
Chapter 15: Lexical Semantics

Pierre Nugues

Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

October 7, 2021



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?



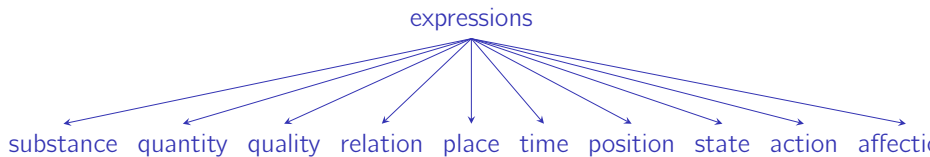
Categories of Words

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



Classes

- Synonymy/Antonymy
- Polysemy
- Hyponyms/Hypernyms `is_a(tree, plant)`, life form, entity
- Meronyms/Holonyms `part_of(leg, table)`
- Grammatical cases: [*nominative* *I*] *broke* [*accusative* *the window*] [*ablative* *with a hammer*]
- Semantic cases: [*actor* *I*] *broke* [*object* *the window*] [*instrument* *with a hammer*]
- Case ambiguity (*The window broke/ I broke the window*)



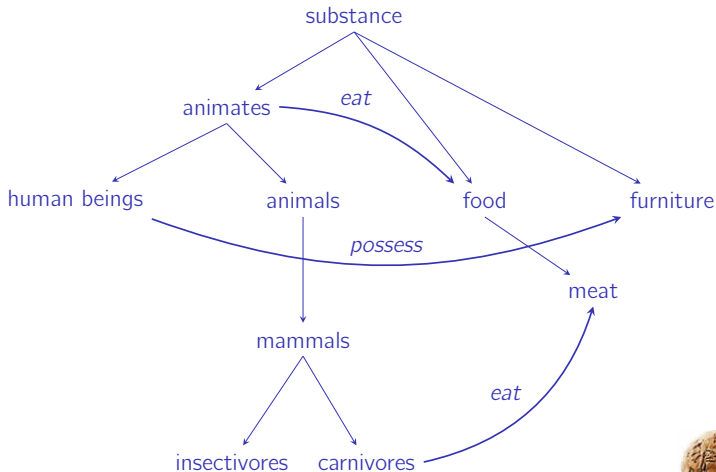
Lexical Database

```
% is_a(?Word, ?Hypernym)
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).

hypernym(X, Y) :- is_a(X, Y).
hypernym(X, Y) :- is_a(X, Z), hypernym(Z, Y).
```



Semantic Networks



An Example: WordNet

| | |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nouns | hyponyms/hypernyms synonyms/antonyms meronyms |
| Adjectives | synonyms/antonyms relational fraternal → 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 |



Semantics and Reasoning

The caterpillar ate the hedgehog.

Representation:

$\exists(X, Y), \text{caterpillar}(X) \wedge \text{hedgehog}(Y) \wedge \text{ate}(X, Y).$

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

| French | German | Danish |
|--------------|-------------|-------------|
| <i>arbre</i> | <i>Baum</i> | |
| | <i>Holz</i> | <i>Træ</i> |
| <i>bois</i> | | |
| <i>forêt</i> | <i>Wald</i> | <i>Skov</i> |

| French | Welsh |
|-------------|---------------|
| | <i>gwyRDD</i> |
| <i>vert</i> | |
| <i>bleu</i> | <i>glas</i> |
| <i>gris</i> | |
| | <i>llwyd</i> |
| <i>brun</i> | |



Sense Tagging Using the Oxford Advanced Learner's Dictionary (OALD)

Sentence: *The patron ordered a meal*

| Words | Definitions | Sense |
|-------------------|------------------------------------------------------------------------------------------------------------------|------------|
| <i>The patron</i> | Correct sense: A customer of a shop, restaurant, theater | 1.2 |
| | Alternate sense: A person who gives money or support to a person, an organization, a cause or an activity | 1.1 |
| <i>ordered</i> | Correct sense: To request somebody to bring food, drink, etc in a hotel, restaurant etc. | 2.3 |
| | Alternate senses: To give an order to somebody | 2.1 |
| | To request somebody to supply or make goods, etc. | 2.2 |
| | To put something in order | 2.4 |
| <i>a meal</i> | Correct sense: The food eaten on such occasion | 1.2 |
| | Alternate sense: An occasion where food is eaten | 1.1 |



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:

$$W_{-m}, W_{-m+1}, \dots, W_{-1}, W, W_1, \dots, W_{m-1}, W_m.$$

If w has n senses, $s_1..s_n$, 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:

$$\begin{aligned} \hat{s} &= \arg \max_{s_i, 1 \leq i \leq n} P(s_i) P(C | s_i), \\ &= \arg \max_{s_i, 1 \leq i \leq n} P(s_i) P(W_{-m}, W_{-m+1}, \dots, W_{-1}, W_1, \dots, W_{m-1}, W_m | s_i). \end{aligned}$$



Naïve Bayes

The Naïve Bayes classifier uses the bag-of-words approach.
We replace

$$P(w_{-m}, w_{-m+1}, \dots, w_{-1}, w_1, \dots, w_{m-1}, w_m | s_i)$$

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



Beyond Words: Predicates and Arguments

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*



Parsing with Cases

The waiter brought the meal to the patron

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

| Case | Type | | Value |
|-----------------------------|---------|--------------|-------------------|
| Agentive | Animate | (Obligatory) | <i>The waiter</i> |
| Objective (or theme) | | (Obligatory) | <i>the meal</i> |
| Dative | Animate | (Optional) | <i>the patron</i> |
| Time | | (Obligatory) | <i>past</i> |



FrameNet

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

- 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].

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

| | | | | | | |
|---------|---------------|----|---------|---------------|----------|------------|
| Sent. 1 | <i>avenge</i> | FE | Avenger | Injured_party | | |
| | | PT | NP | NP | | |
| | | GF | Ext | Object | | |
| Sent. 2 | <i>avenge</i> | FE | Avenger | Injury | | |
| | | PT | NP | NP | | |
| | | GF | Ext | Obj | | |
| Sent. 3 | <i>avenge</i> | FE | Avenger | Injured_party | Offender | Punishment |
| | | PT | NP | NP | PP | PPing |
| | | GF | Ext | Obj | Comp | Comp |



Automatic Frame-semantic Analysis (Johansson, 2008)

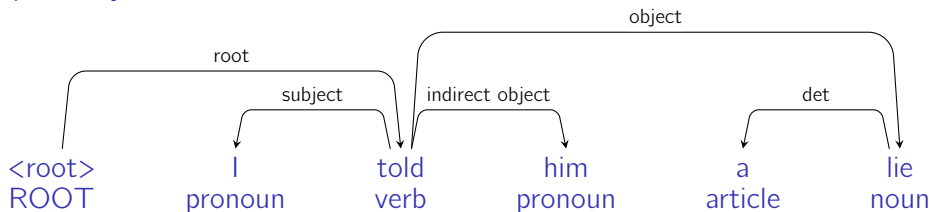
Given a sentence:

*I **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:



Classification of Semantic Arguments (Johansson, 2008)

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.

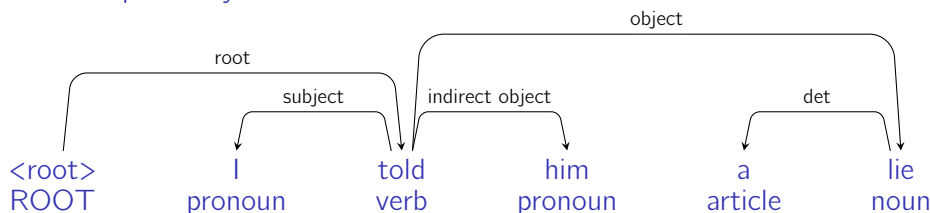
What features are useful for this task? Examples:

- 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



Feature Extraction (Johansson, 2008)

Given a dependency tree:



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

| Word | Grammatical function | Voice | Argument |
|------------|----------------------|--------|-------------------------|
| <i>I</i> | Subject | Active | <i>speaker</i> (Arg0) |
| <i>him</i> | Indirect object | Active | <i>hearer</i> (Arg2) |
| <i>lie</i> | Direct object | Active | <i>utterance</i> (Arg1) |



Propbank

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:

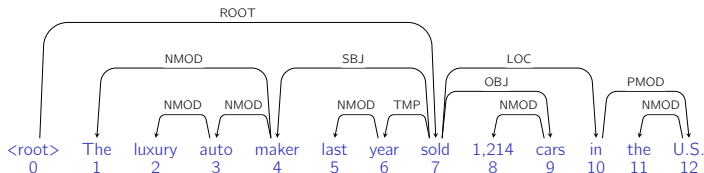
The luxury auto maker last year sold 1,214 cars in the U.S.

| ID | Form | Lemma | POS | Feats | Head | Deprel | FillPred | Sense | APred1 | APred2 |
|----|--------|--------|-----|-------|------|--------|----------|----------|--------|--------|
| 1 | The | the | DT | — | 4 | NMOD | — | — | — | — |
| 2 | luxury | luxury | NN | — | 3 | NMOD | — | — | A1 | — |
| 3 | auto | auto | NN | — | 4 | NMOD | — | — | A1 | — |
| 4 | maker | maker | NN | — | 7 | SBJ | Y | maker.01 | A0 | A0 |
| 5 | last | last | JJ | — | 6 | NMOD | — | — | — | — |
| 6 | year | year | NN | — | 7 | TMP | — | — | — | AM-TMP |
| 7 | sold | sell | VBD | — | 0 | ROOT | Y | sell.01 | — | — |
| 8 | 1,214 | 1,214 | CD | — | 9 | NMOD | — | — | — | — |
| 9 | cars | car | NNS | — | 7 | OBJ | — | — | — | A1 |
| 10 | in | in | IN | — | 7 | LOC | — | — | — | AM-LOC |
| 11 | the | the | DT | — | 12 | NMOD | — | — | — | — |
| 12 | U.S. | u.s. | NNP | — | 10 | PMOD | — | — | — | — |

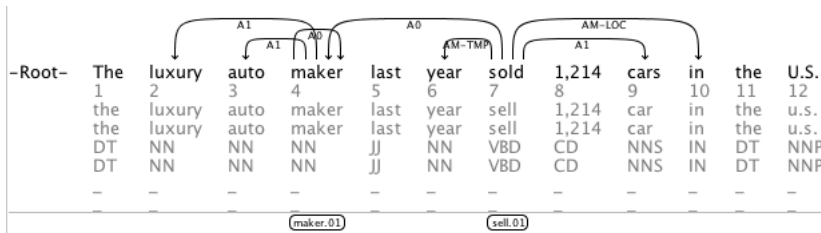


Visualizing Dependencies

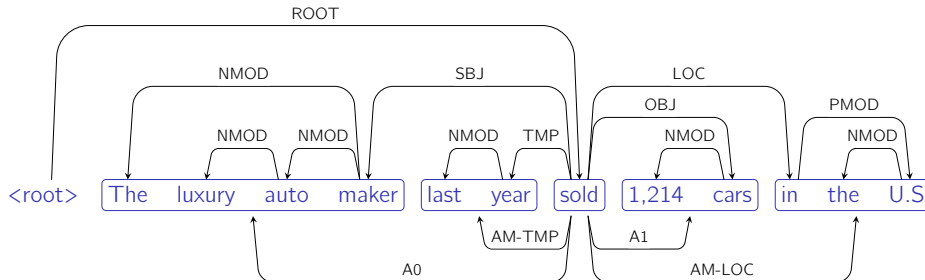
Syntactic dependencies:



Semantic dependencies (predicate–argument structures):



Alternative Visualization



| | The | luxury | auto | maker | last | year | sold | 1,214 | cars | in | the | U.S. |
|----------|-----|--------|------|-------|--------|------|------|-------|------|--------|-----|------|
| maker.01 | | A1 | | A0 | | | | | | | | |
| sell.01 | A0 | | | | AM-TMP | | | A1 | | AM-LOC | | |

The Real Annotation

The participants are given the predicted values up to the semantic columns so that they can focus on the semantic step.

For the sentence:

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

The input is in bold.

| ID | Form | Lemma | PLemma | POS | PPOS | Feats | PFeats | Head | PHead | Deprel | PDeprel | FillPred | Sense | APred1 | APred2 |
|----|--------|--------|--------|-----|------|-------|--------|------|-------|--------|---------|----------|----------|--------|--------|
| 1 | The | the | the | DT | DT | — | — | 4 | 4 | NMOD | NMOD | — | — | — | — |
| 2 | luxury | luxury | luxury | NN | NN | — | — | 3 | 3 | NMOD | NMOD | — | — | A1 | — |
| 3 | auto | auto | auto | NN | NN | — | — | 4 | 4 | NMOD | NMOD | — | — | A1 | — |
| 4 | maker | maker | maker | NN | NN | — | — | 7 | 7 | SBJ | SBJ | Y | maker.01 | A0 | A0 |
| 5 | last | last | last | JJ | JJ | — | — | 6 | 6 | NMOD | NMOD | — | — | — | — |
| 6 | year | year | year | NN | NN | — | — | 7 | 7 | TMP | TMP | — | — | — | AM-TMP |
| 7 | sold | sell | sell | VBD | VBD | — | — | 0 | 0 | ROOT | ROOT | Y | sell.01 | — | — |
| 8 | 1,214 | 1,214 | 1,214 | CD | CD | — | — | 9 | 9 | NMOD | NMOD | — | — | — | — |
| 9 | cars | car | car | NNS | NNS | — | — | 7 | 7 | OBJ | OBJ | — | — | — | A1 |
| 10 | in | in | in | IN | IN | — | — | 7 | 7 | LOC | LOC | — | — | — | AM-LOC |
| 11 | the | the | the | DT | DT | — | — | 12 | 12 | NMOD | NMOD | — | — | — | — |
| 12 | U.S. | u.s. | u.s. | NNP | NNP | — | — | 10 | 10 | PMOD | PMOD | — | — | — | — |



Parsing Pipeline

Input sentence

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.??)

(sell.??)

Predicate sense disambiguation

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

(maker.01)

(sell.01)

Argument identification

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

(sell.01)

Argument labeling

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

A0

AM-TMP

(sell.01)

A1

AM-LOC



Semantic Parsing As a Tagging Operation

We can also apply a technique similar to that in chunking (Zhou and Xu, 2015):

Starting from the segments:

| | | | | | | | | | | | | |
|----------|-----|--------|------|-------|--------|------|------|-------|------|--------|-----|------|
| | The | luxury | auto | maker | last | year | sold | 1,214 | cars | in | the | U.S. |
| maker.01 | | A1 | | A0 | | | | | | | | |
| sell.01 | A0 | | | | AM-TMP | | | A1 | | AM-LOC | | |

We annotate the arguments with the IOB2 tagset (Begin, Inside, Outside):

| | The | luxury | auto | maker | last | year | sold | 1,214 | cars | in | the | U.S. |
|----------|--------|--------|--------|--------|-------|-------|------|--------|--------|-------|-------|-------|
| maker.01 | O | B-ARG1 | I-ARG1 | B-ARG0 | O | O | O | O | O | O | O | O |
| sell.01 | B-ARG0 | I-ARG0 | I-ARG0 | I-ARG0 | B-TMP | I-TMP | B-V | B-ARG1 | I-ARG1 | B-LOC | I-LOC | I-LOC |



Semantic Parsing as a Tagging Operation (II)

The annotated corpus:

| | The | luxury | auto | maker | last | year | sold | 1,214 | cars | in | the | U.S. |
|----------|--------|--------|--------|--------|-------|-------|------|--------|--------|-------|-------|-------|
| maker.01 | O | B-ARG1 | I-ARG1 | B-ARG0 | O | O | O | O | O | O | O | O |
| sell.01 | B-ARG0 | I-ARG0 | I-ARG0 | I-ARG0 | B-TMP | I-TMP | B-V | B-ARG1 | I-ARG1 | B-LOC | I-LOC | I-LOC |

Collecting the features from Zhou and Xu (2015):

- ① The input is the word sequence and the output is the tag sequence: sequence-to-sequence learning;
- ② The features are similar to those used for chunking:
 - The current word;
 - The predicate (from a previous detection);
 - The predicate context (three words centered on the predicate);
 - if the current word is in the predicate context;
- ③ The process is repeated as many times as there are predicates in the sentence.



Semantic Parsing as a Tagging Operation (III)

The annotated corpus:

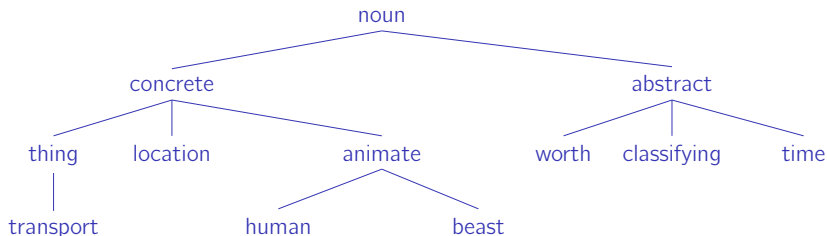
| | The | luxury | auto | maker | last | year | sold | 1,214 | cars | in | the | U.S. |
|----------|--------|--------|--------|--------|-------|-------|------|--------|--------|-------|-------|-------|
| maker.01 | O | B-ARG1 | I-ARG1 | B-ARG0 | O | O | O | O | O | O | O | O |
| sell.01 | B-ARG0 | I-ARG0 | I-ARG0 | I-ARG0 | B-TMP | I-TMP | B-V | B-ARG1 | I-ARG1 | B-LOC | I-LOC | I-LOC |

$$\mathbf{X} = \begin{bmatrix} \text{The} & \text{sell.01} & \text{year sold 1,214} & 0 \\ \text{luxury} & \text{sell.01} & \text{year sold 1,214} & 0 \\ \text{auto} & \text{sell.01} & \text{year sold 1,214} & 0 \\ \text{maker} & \text{sell.01} & \text{year sold 1,214} & 0 \\ \dots & \dots & \dots & \dots \\ \text{The} & \text{maker.01} & \text{auto maker last} & 0 \\ \text{luxury} & \text{maker.01} & \text{auto maker last} & 0 \\ \text{luxury} & \text{maker.01} & \text{auto maker last} & 0 \\ \text{auto} & \text{maker.01} & \text{auto maker last} & 1 \\ \dots & \dots & \dots & \dots \end{bmatrix} ; \mathbf{y} = \begin{bmatrix} \text{B-ARG0} \\ \text{I-ARG0} \\ \text{I-ARG0} \\ \text{I-ARG0} \\ \dots \\ \text{O} \\ \text{B-ARG1} \\ \text{I-ARG1} \\ \text{B-ARG0} \\ \dots \end{bmatrix}$$



EVAR

EVAR is a German project that aims at providing information on trains



EVAR's Case Grammar

- ❶ 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
- ❷ 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
- ❸ 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**

[<Impactor> The rock] HIT [<Impactee> the sand] with a thump

Source: <http://framenet.icsi.berkeley.edu/>

In Carsim:

[ACTOR En personbil] körde [TIME vid femtiden] [TIME på torsdagseftermiddagen] in [VICTIM i ett radhus] [LOC i ett äldreboende] [LOC på Alvägen] [LOC i Enebyberg] [LOC norr om Stockholm].

