Language Technology
http://cs.lth.se/edan20/
Chapter 15: Semantics

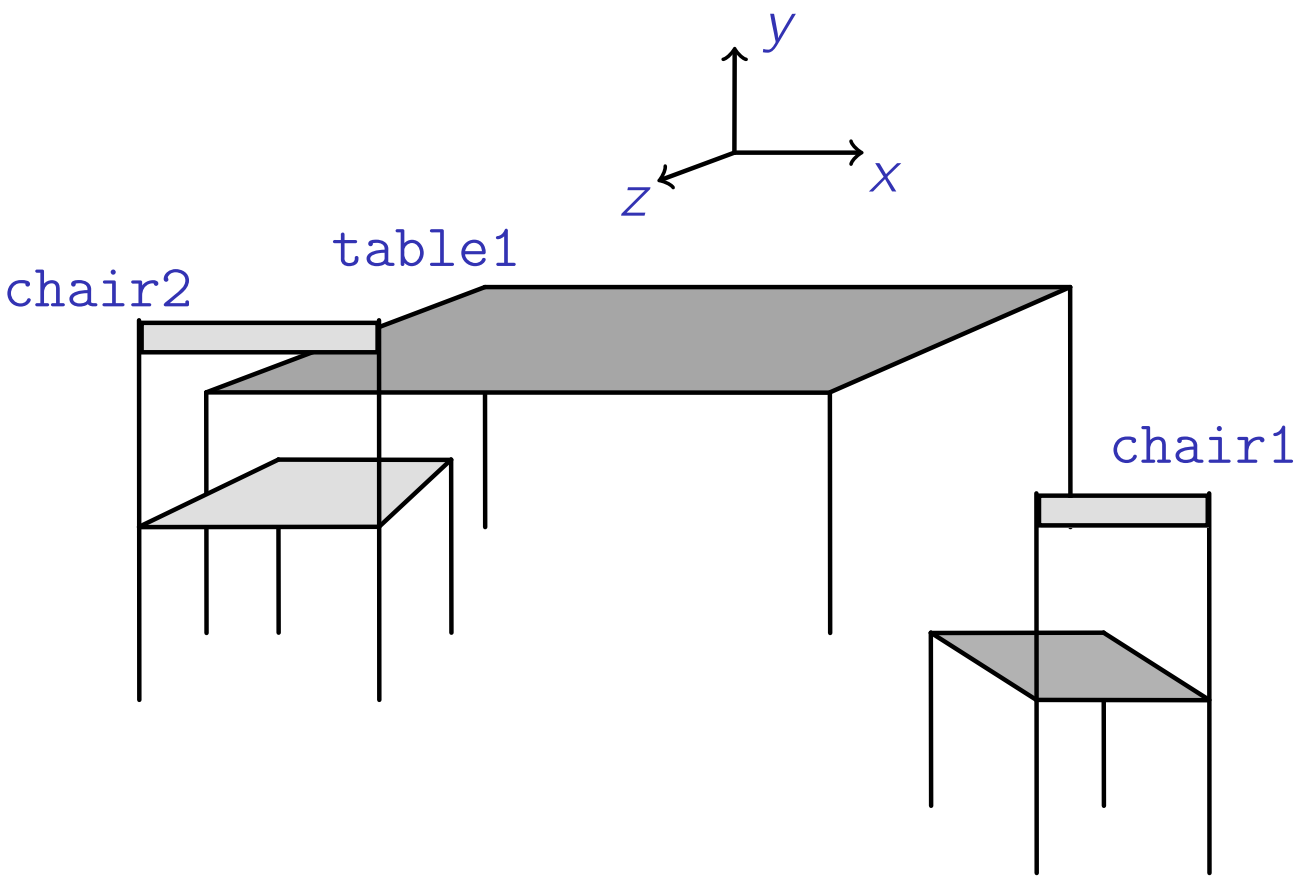
Pierre Nugues

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

October 13, 2022

The State of Affairs

Two people at a table, Pierre and Socrates, and a robot waiter.



Formal Semantics

Its goal is to:

- Represent the state of affairs.
- Translate phrases or sentences such as *The robot brought the meal* or *the meal on the table* into logic formulas
- Solve references: Link words to real entities
- Reason about the world and the sentences.

A way to represent things and relations is to use first-order predicate calculus (FOPC) and predicate–argument structures

Predicates

Constants:

```
% The people:
  'Socrates'.
  'Pierre'.

% The chairs:
  chair1.     % chair #1
  chair2.     % chair #2

% The unique table:
  table1.     % table #1
```

Predicates to encode properties:

```
person('Pierre').
person('Socrates').

object(table1).
object(chair1).
object(chair2).

chair(chair1).
chair(chair2).
table(table1).
```

Predicates to encode relations:

```
in_front_of(chair1, table1).
on('Pierre', table1).
```

Logical Forms

Logical forms map sentences onto predicate-argument structures

I would like to book a late flight to Boston

```
would(like_to(i,
   book(i,
       np_pp(a(late(flight)),
           X^to(X, boston)))))
```

This enables us to represent knowledge and build a knowledge graph.

RDF and SPARQL

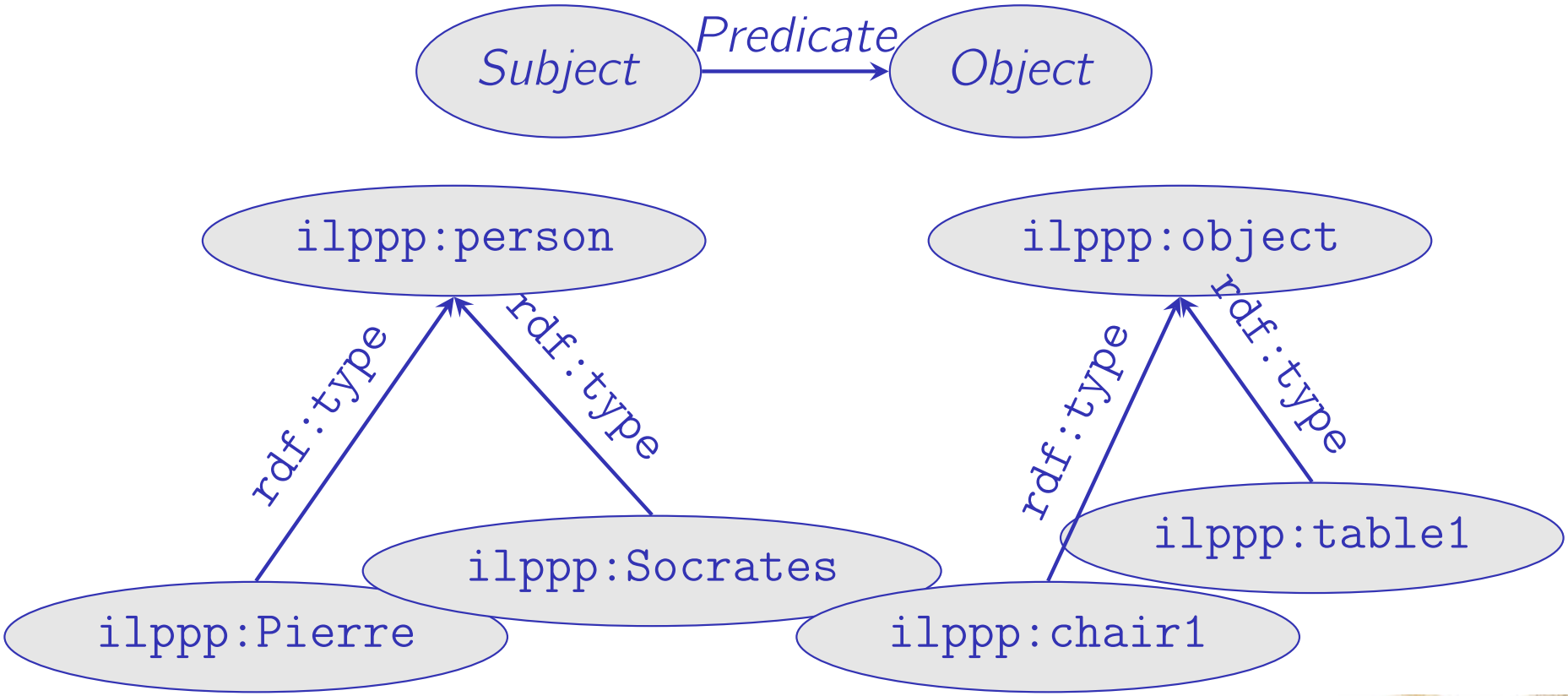
RDF: A popular graph format to encode knowledge.
SPARQL: A query language for RDF
In many ways, very similar to Prolog.

```
ilppp:Pierre rdf:type ilppp:person.
ilppp:Socrates rdf:type ilppp:person.

ilppp:table1 rdf:type ilppp:object.
ilppp:chair1 rdf:type ilppp:object.
ilppp:chair2 rdf:type ilppp:object.

ilppp:chair1 ilppp:in_front_of ilppp:table1.
ilppp:Pierre ilppp:on ilppp:table1.
```

RDF Triples


Subject
Predicate
Object
ilppp:person
ilppp:object
rdf:type
rdf:type
rdf:type
rdf:type
ilppp:Socrates
ilppp:table1
ilppp:Pierre
ilppp:chair1

RDF and SPARQL

Prolog:

```
?- object(X), object(Y), in_front_of(X, Y).
X = chair1,
Y = table1.
```

SPARQL:

```
SELECT ?x ?y
WHERE
{
  ?x rdf:type ilppp:object.
  ?y rdf:type ilppp:object.
  ?x ilppp:in_front_of ?y
}
```

Variables	?x	?y
Values	ilppp:chair1	ilppp:table1

DBpedia, Yago, Wikidata, and Freebase

Graph databases consisting of billions of RDF triples.
Coming from a variety of sources such as Wikipedia infoboxes:

```
{{Infobox settlement
| official_name          = Busan Metropolitan City
...
| area_total_km2         = 767.35
...
| population_total       = 3,525,913
...
}}
```

DBpedia: The result of a systematic triple extraction from infoboxes

```
dbpedia:Busan foaf:name "Busan Metropolitan City"@en .
dbpedia:Busan dbo:populationTotal "3525913".
dbpedia:Busan dbo:areaTotal "7.6735E8" .
...
```

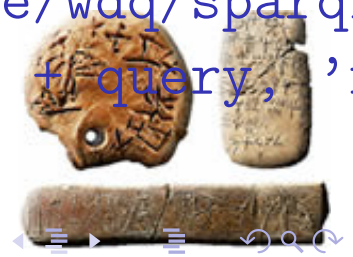
Wikidata

Wikidata provides a SPQRL endpoint based on Wikipedia data:
https://query.wikidata.org

```
query = '''
SELECT ?entity ?population
WHERE
{
  ?entity rdfs:label "Busan"@en .
  ?entity wdt:P1082 ?population.

}'''

url = 'https://query.wikidata.org/bigdata/namespace/wdq/sparql
data = requests.get(url, params={'query': prefixes + query, 'f
```



Semantics and the Real World: Words

Techniques to map a sentence to a logical form:

- First attempts used logic
- Dominant approaches use lexical semantics, starting with words.

Issues:

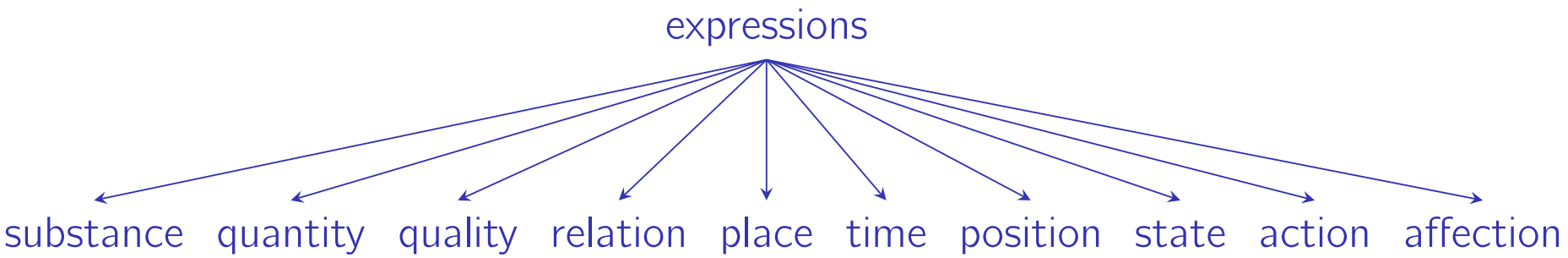
- Classes of words: If it is hot, can it be cold?
- Definitions: What is a meal? What is table?
- Reasoning (or entailment): The meal is on the table. Is it cold?

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)

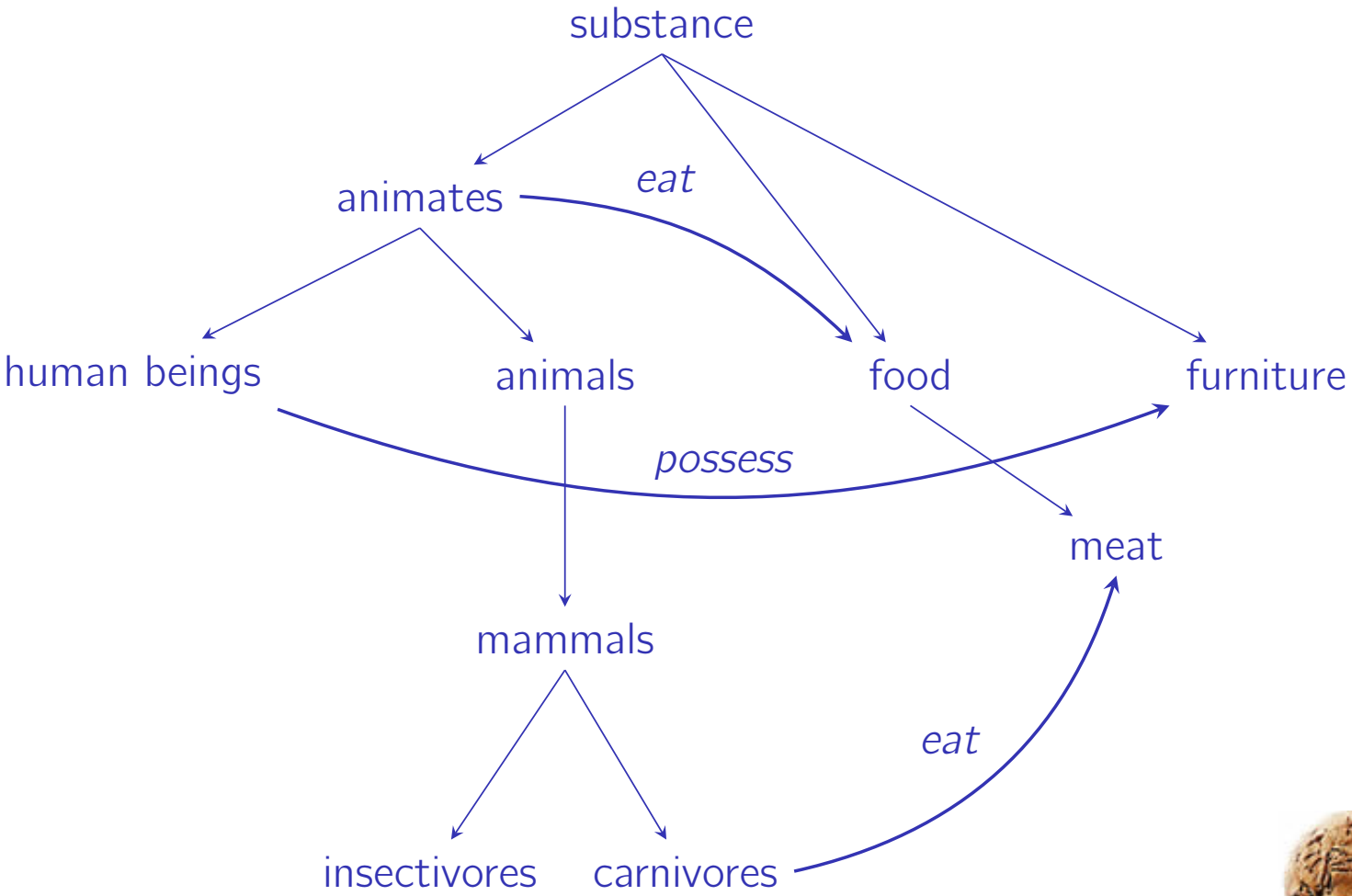
Representation of Categories



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*)

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

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
arbre	*Baum*	
	Holz	*Træ*
bois		
forêt	*Wald*	*Skov*

French	Welsh
	gwyrdd
vert	
bleu	*glas*
gris	
	llwyd
brun	

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

Sentence: *The patron ordered a meal*

Words	Definitions	Sense
The patron	**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**
ordered	**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**
a meal	**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:

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

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

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

Using Bayes' rule, we have:

$$\begin{aligned} \hat{s} &= \underset{s_i, 1 \leq i \leq n}{\arg\max} P(s_i)P(C|s_i), \\ &= \underset{s_i, 1 \leq i \leq n}{\arg\max} P(s_i)P(w_{-m}, w_{-m+1}, \ldots, w_{-1}, w_1, \ldots, w_{m-1}, w_m|s_i). \end{aligned}$$

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	Object = *door*
John opened the door	Object = *door* and Agent = *John*
The wind opened the door	Object = *door* and Agent = *wind*
John opened the door with a chisel	Object = *door*, Agent = *John*, and Instrument = *chisel*

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

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.

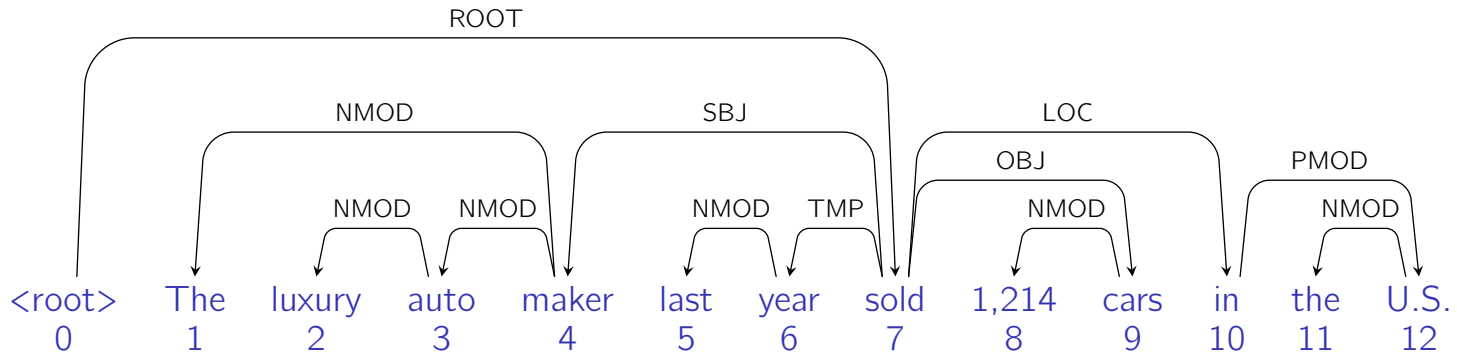
CoNLL annotation format of the sentence:

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

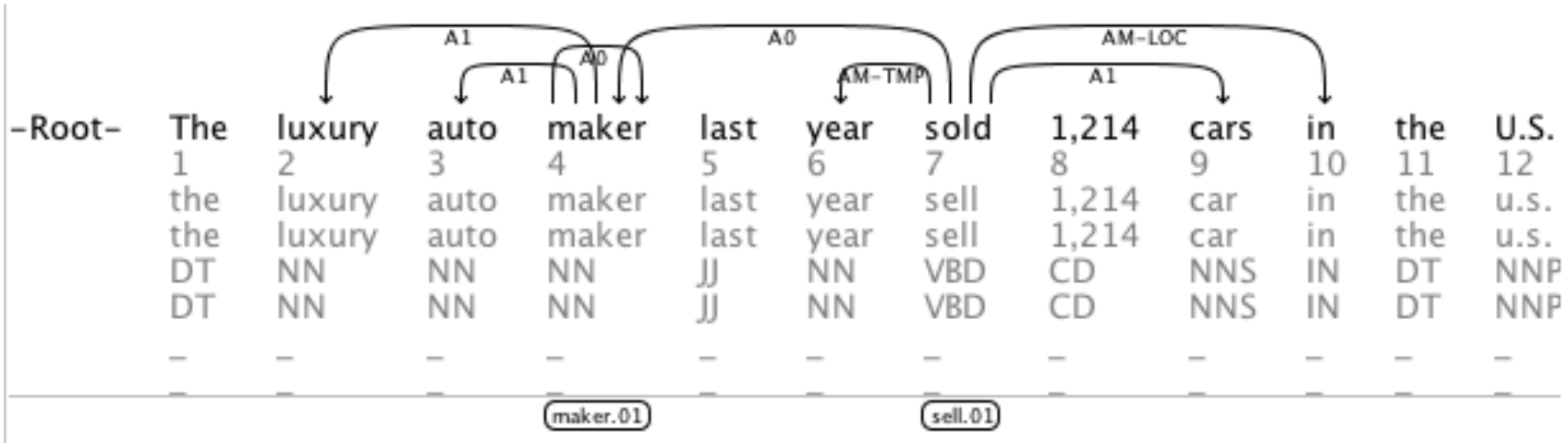
ID	Form	Lemma	POS	Feats	Head	Deprel	FillPred	Sense	APred1	APred2
1	The	the	DT	_	4	NMOD	_	_	_	_
2	luxury	luxury	NN	_	3	NMOD	_	_	A1	_
3	auto	auto	NN	_	4	NMOD	_	_	A1	_
4	maker	maker	NN	_	7	SBJ	Y	maker.01	A0	A0
5	last	last	JJ	_	6	NMOD	_	_	_	_
6	year	year	NN	_	7	TMP	_	_	_	AM-TMP
7	sold	sell	VBD	_	0	ROOT	Y	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

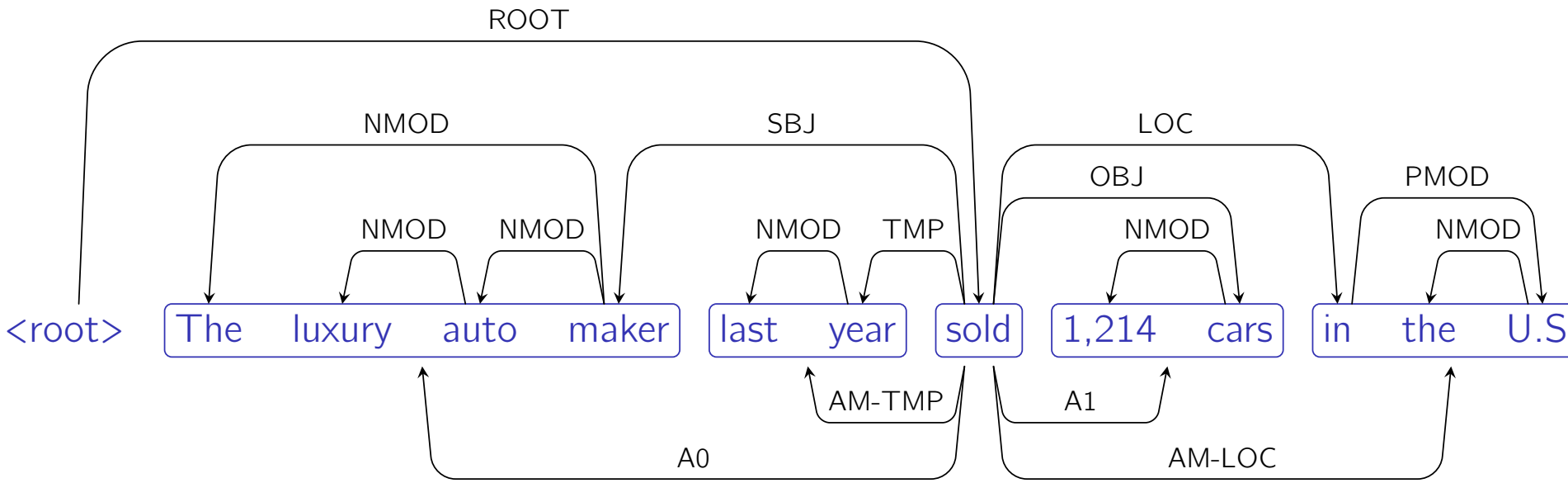
Syntactic dependencies:



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		

Parsing Pipeline (Old Style)

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 sell.01 A1

AM-TMP AM-LOC

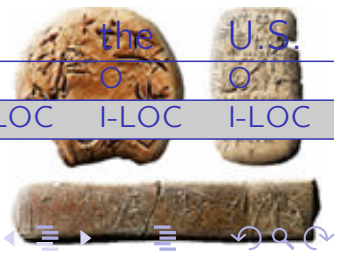
Semantic Parsing As a Tagging Operation

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.

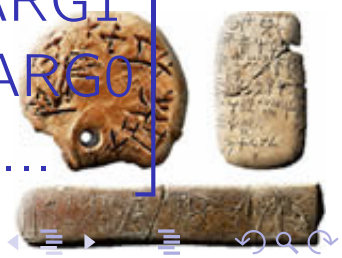
LSTMs and transformers yield even better results

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

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



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].