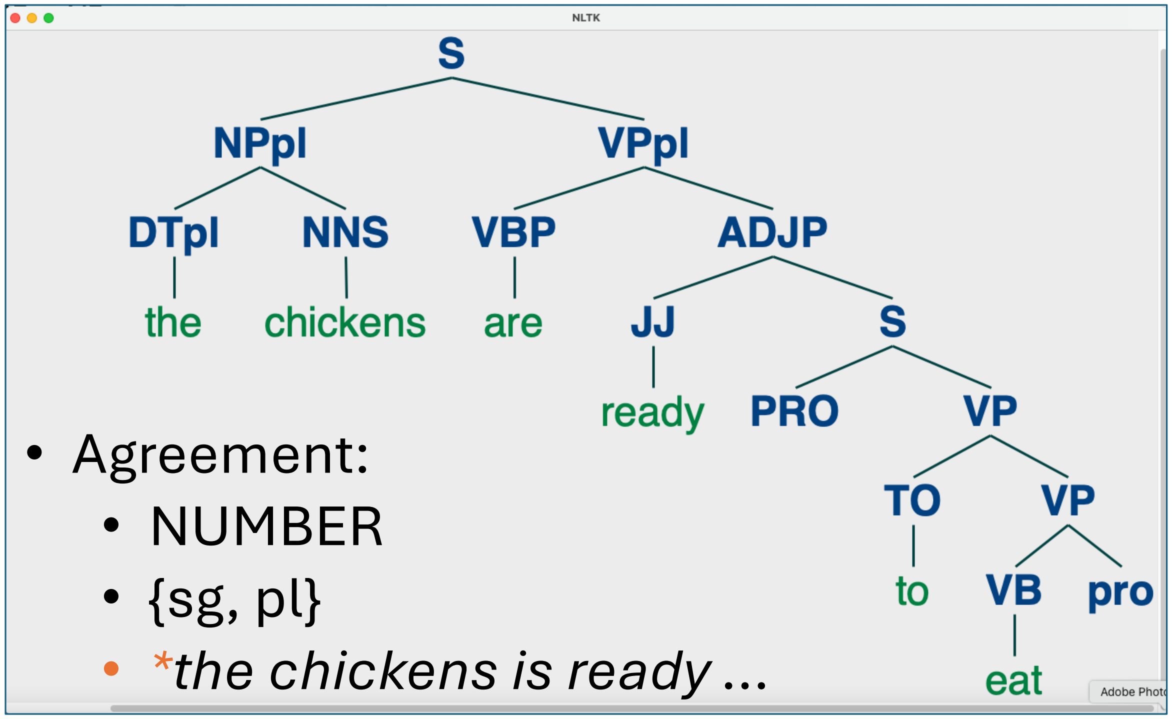
LING 388: Computers and Language

Lecture 28

Last Time

- Syntax: talked about nltk (context-free) grammar rules
  - ability to specify empty categories
  - inability to handle Control of PRO (empty subject pronoun of nonfinite clauses)
  - *there's more but we will move on to a new topic today*



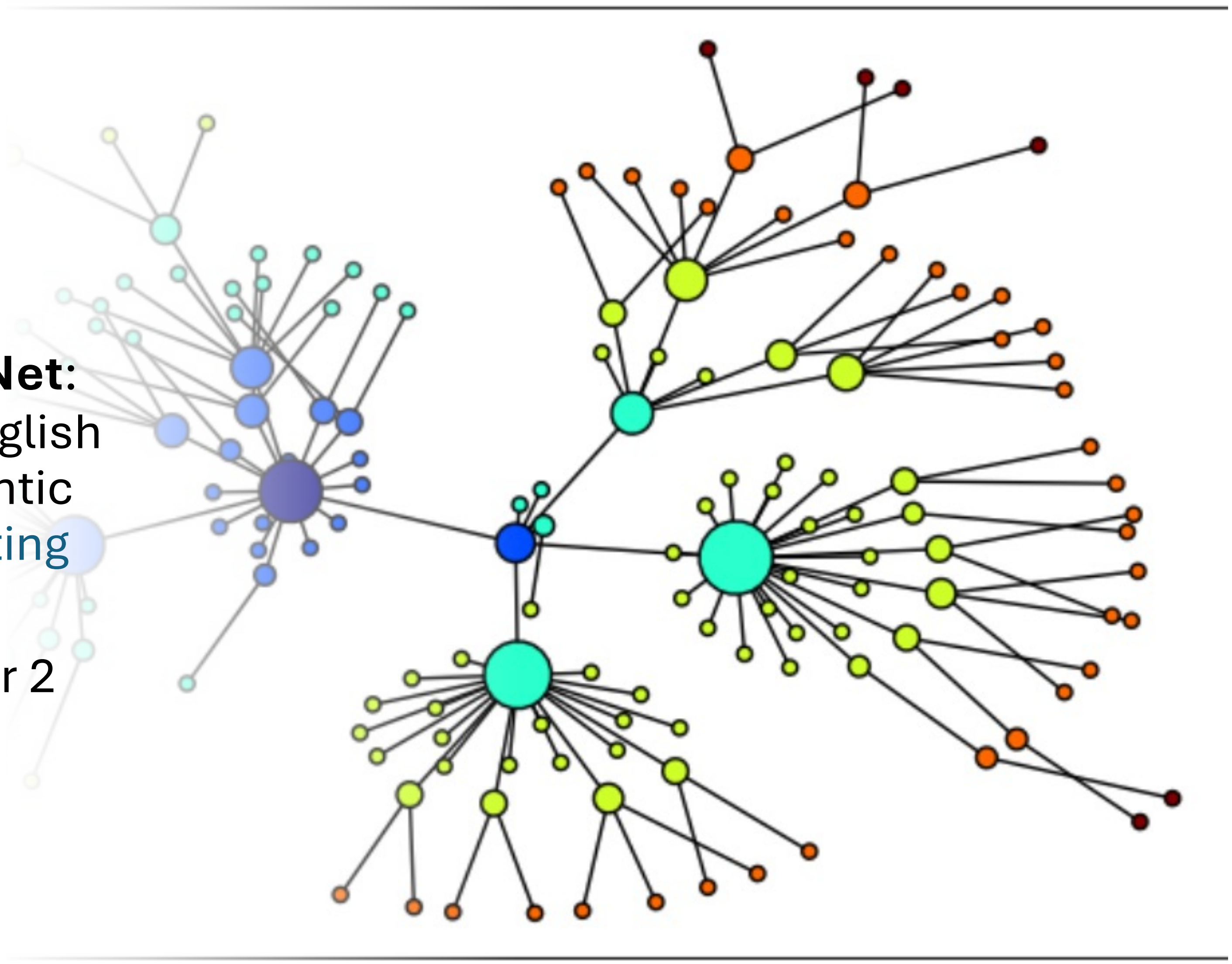
- Agreement:
  - NUMBER
  - {sg, pl}
  - **the chickens is ready …*

Today's Topic

- *Fill out the course surveys please*
- WordNet:
  - a freely-available dictionary of English word senses organized by synonym (sets).
  - not for prepositional senses, e.g. *with, from, in* etc.
  - for **open class** words: nouns, verbs, adverbs and adjectives
  - can use it online (web interface) and inside `nltk`
  - *next time, last lecture: we'll relate WordNet to word embeddings*

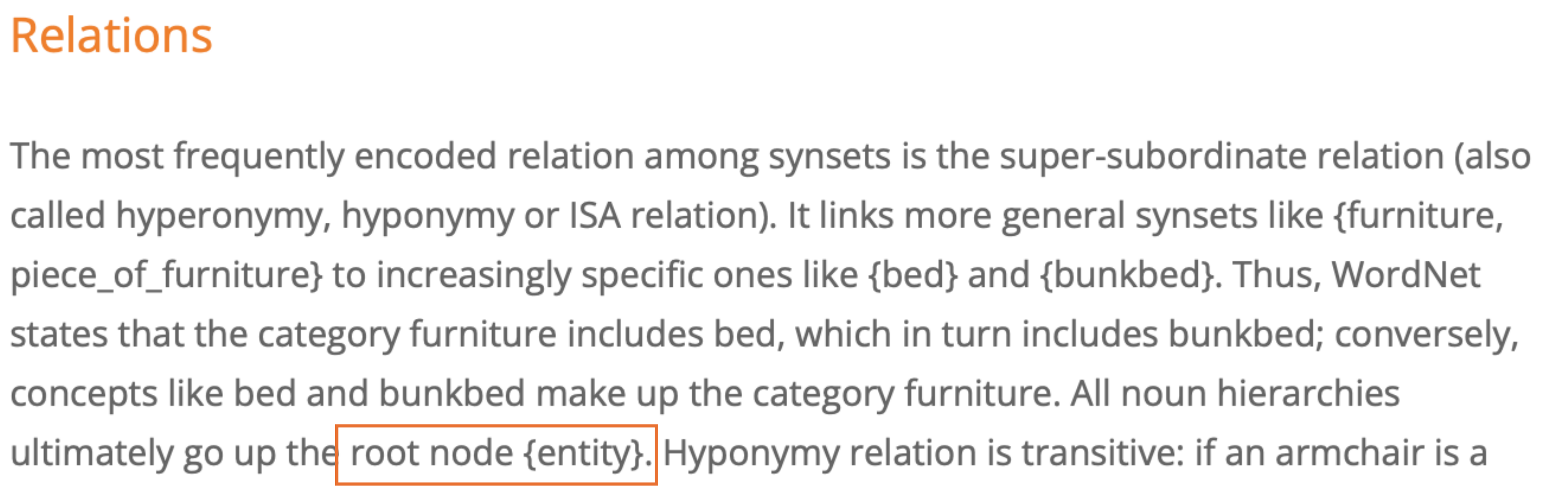
WordNet

- **Princeton WordNet**: a dictionary of English words with semantic relations connecting word senses
- nltk book: chapter 2 and 4



WordNet

- Relations between word senses grouped into synonym sets (**synsets**)



Relations

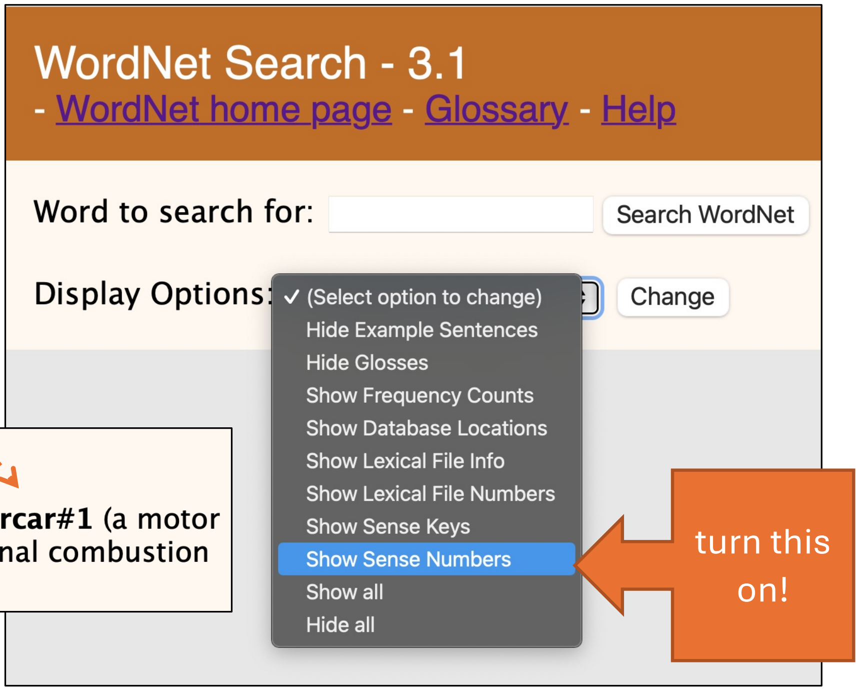
The most frequently encoded relation among synsets is the super-subordinate relation (also called hyperonymy, hyponymy or ISA relation). It links more general synsets like {furniture, piece_of_furniture} to increasingly specific ones like {bed} and {bunkbed}. Thus, WordNet states that the category furniture includes bed, which in turn includes bunkbed; conversely, concepts like bed and bunkbed make up the category furniture. All noun hierarchies ultimately go up the root node {entity}. Hyponymy relation is transitive: if an armchair is a

nltk book: **2.5.1 Senses and Synonyms**

Synonyms:

- Benz is credited with the invention of the motorcar.
- Benz is credited with the invention of the automobile.

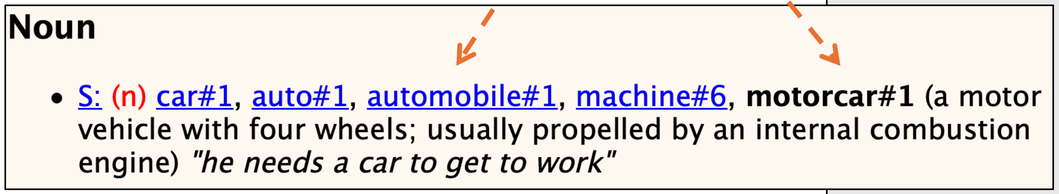
- http://wordnetweb.princeton.edu/perl/webwn



WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: [ ] Search WordNet

Display Options: Change

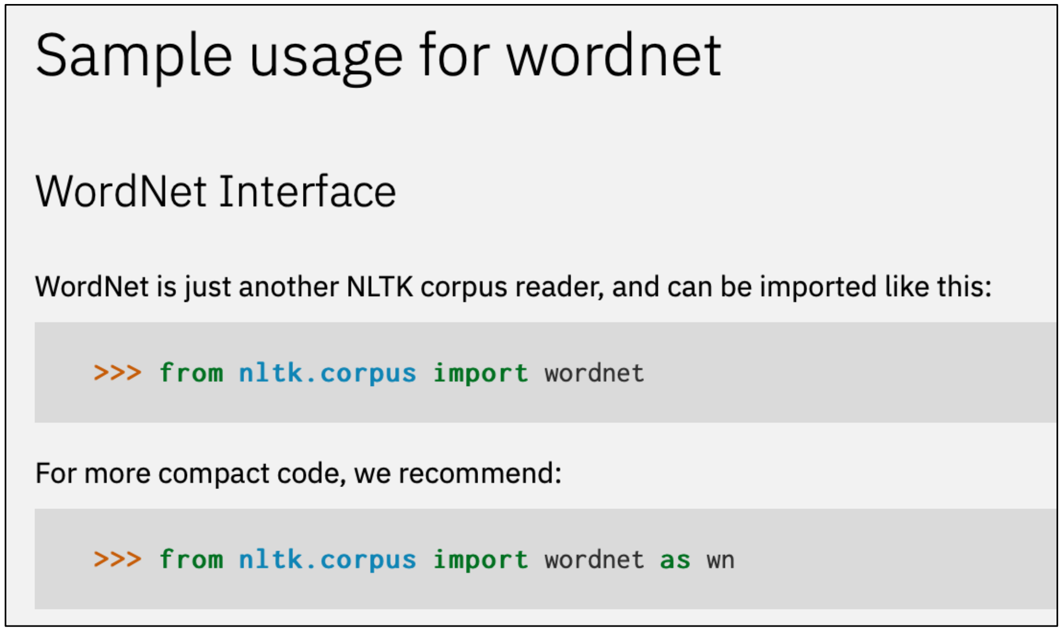


Noun

- S: (n) car#1, auto#1, automobile#1, machine#6, **motorcar#1** (a motor vehicle with four wheels; usually propelled by an internal combustion engine) *"he needs a car to get to work"*

nltk and WordNet

http://www.nltk.org/howto/wordnet.html



Sample usage for wordnet

WordNet Interface

WordNet is just another NLTK corpus reader, and can be imported like this:

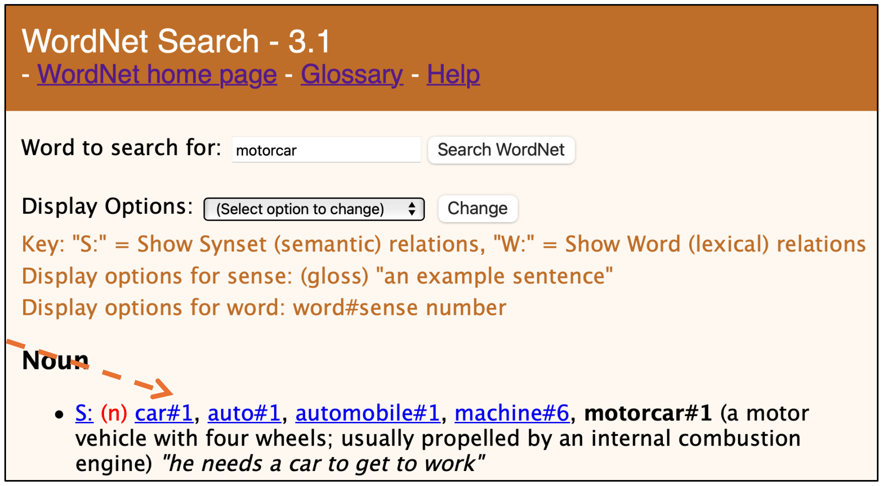
```
>>> from nltk.corpus import wordnet
```

For more compact code, we recommend:

```
>>> from nltk.corpus import wordnet as wn
```

nltk book: **2.5.1 Senses and Synonyms**

```
>>> from nltk.corpus import wordnet as wn
>>> wn.synsets('motorcar')
[Synset('car.n.01')]
>>> s = wn.synset('car.n.1')
>>> s
Synset('car.n.01')
>>> s.lemmas()
[Lemma('car.n.01.car'), Lemma('car.n.01.auto'),
Lemma('car.n.01.automobile'),
Lemma('car.n.01.machine'),
Lemma('car.n.01.motorcar')]
>>> s.lemma_names()
['car', 'auto', 'automobile', 'machine',
'motorcar']
>>> s.definition()
'a motor vehicle with four wheels; usually
propelled by an internal combustion engine'
>>> s.examples()
['he needs a car to get to work']
```



WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: motorcar Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Display options for word: word#sense number

Noun

- S: (n) car#1, auto#1, automobile#1, machine#6, **motorcar#1** (a motor vehicle with four wheels; usually propelled by an internal combustion engine) *"he needs a car to get to work"*

Key:
- **synset** = synonym set
- **lemma** = word + sense number (member of synset)
- (semantic) **relation** = link between synset

nltk WordNet Notation

A **synset** is uniquely identified with a 3-part name of the form:
`word.pos.nn`

- word = "head" of the synset is the first **lemma** listed
- pos = [asrnv]   (part of speech: adjective/satellite/adverb/noun/verb)

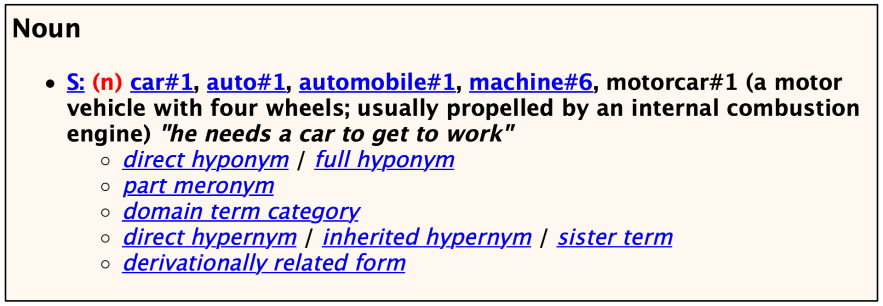
```
wn.synsets('dog')
```

```
[Synset('dog.n.01'), Synset('frump.n.01'), Synset('dog.n.03'),
Synset('cad.n.01'), Synset('frank.n.02'), Synset('pawl.n.01'),
Synset('andiron.n.01'), Synset('chase.v.01')]
```

```
wn.synsets('animal')
```

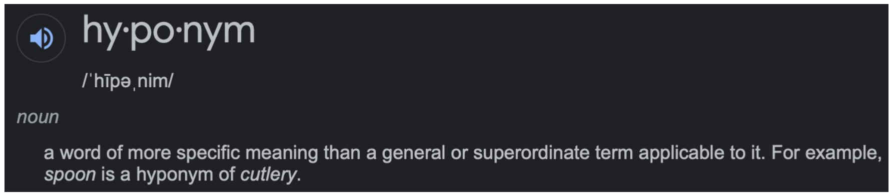
```
[Synset('animal.n.01'), Synset('animal.s.01')]
```

nltk book: **2.5.2 The WordNet Hierarchy**



Noun

- **S: (n) car#1, auto#1, automobile#1, machine#6, motorcar#1 (a motor vehicle with four wheels; usually propelled by an internal combustion engine) *"he needs a car to get to work"***
  - *direct hyponym* / *full hyponym*
  - *part meronym*
  - *domain term category*
  - *direct hypernym* / *inherited hypernym* / *sister term*
  - *derivationally related form*

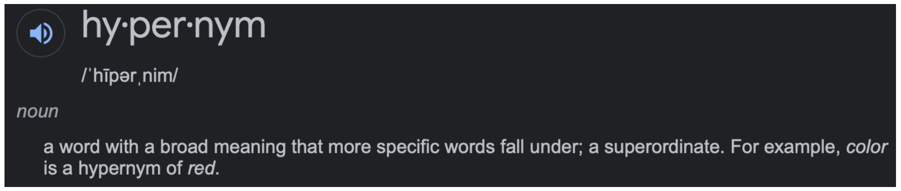


hy·po·nym

/ˈhīpəˌnim/

noun

a word of more specific meaning than a general or superordinate term applicable to it. For example, *spoon* is a hyponym of *cutlery*.

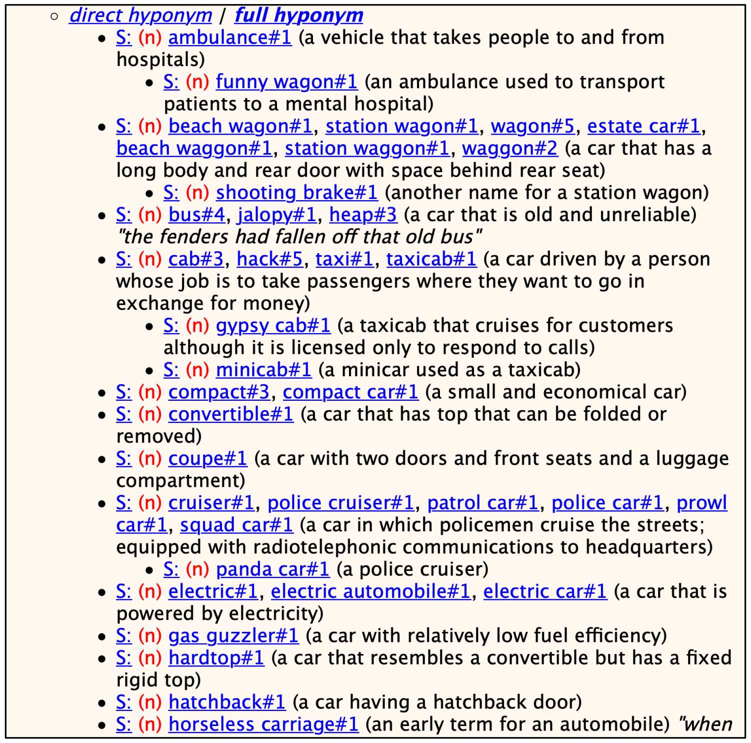


hy·per·nym

/ˈhīpərˌnim/

noun

a word with a broad meaning that more specific words fall under; a superordinate. For example, *color* is a hypernym of *red*.



- *direct hyponym* / ***full hyponym***
  - S: (n) ambulance#1 (a vehicle that takes people to and from hospitals)
    - S: (n) funny wagon#1 (an ambulance used to transport patients to a mental hospital)
  - S: (n) beach wagon#1, station wagon#1, wagon#5, estate car#1, beach waggon#1, station waggon#1, waggon#2 (a car that has a long body and rear door with space behind rear seat)
    - S: (n) shooting brake#1 (another name for a station wagon)
  - S: (n) bus#4, jalopy#1, heap#3 (a car that is old and unreliable) *"the fenders had fallen off that old bus"*
  - S: (n) cab#3, hack#5, taxi#1, taxicab#1 (a car driven by a person whose job is to take passengers where they want to go in exchange for money)
    - S: (n) gypsy cab#1 (a taxicab that cruises for customers although it is licensed only to respond to calls)
    - S: (n) minicab#1 (a minicar used as a taxicab)
  - S: (n) compact#3, compact car#1 (a small and economical car)
  - S: (n) convertible#1 (a car that has top that can be folded or removed)
  - S: (n) coupe#1 (a car with two doors and front seats and a luggage compartment)
  - S: (n) cruiser#1, police cruiser#1, patrol car#1, police car#1, prowl car#1, squad car#1 (a car in which policemen cruise the streets; equipped with radiotelephonic communications to headquarters)
    - S: (n) panda car#1 (a police cruiser)
  - S: (n) electric#1, electric automobile#1, electric car#1 (a car that is powered by electricity)
  - S: (n) gas guzzler#1 (a car with relatively low fuel efficiency)
  - S: (n) hardtop#1 (a car that resembles a convertible but has a fixed rigid top)
  - S: (n) hatchback#1 (a car having a hatchback door)
  - S: (n) horseless carriage#1 (an early term for an automobile) *"when*

nltk book: **2.5.2 The WordNet Hierarchy**

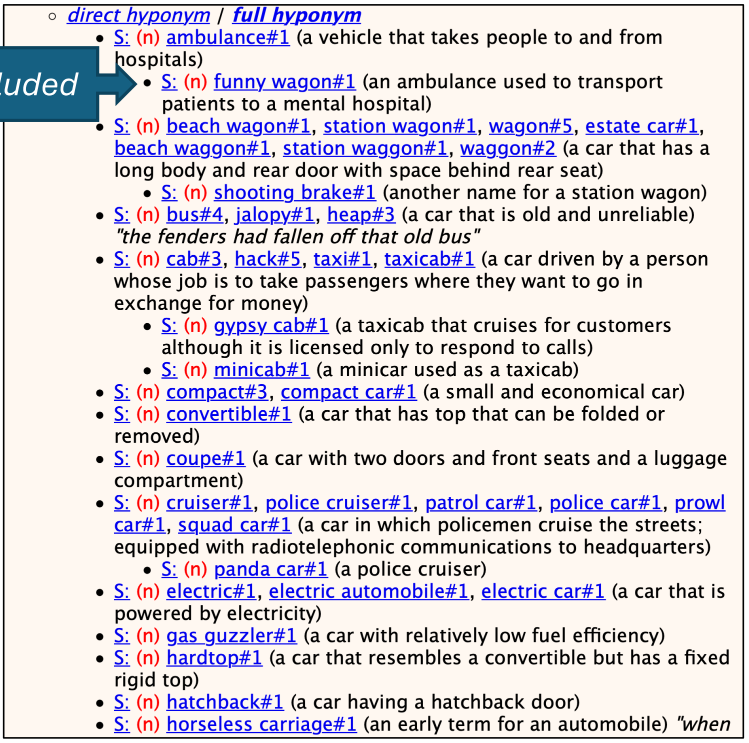
```
s = wn.synset('car.n.1')

>>> [lemma.name() for synset in s.hyponyms()
for lemma in synset.lemmas()]

['ambulance', 'beach_wagon', 'station_wagon',
'wagon', 'estate_car', 'beach_waggon',
'station_waggon', 'waggon', 'bus', 'jalopy',
'heap', 'cab', 'hack', 'taxi', 'taxicab',
'compact', 'compact_car', 'convertible',
'coupe', 'cruiser', 'police_cruiser',
'patrol_car', 'police_car', 'prowl_car',
'squad_car', 'electric',
'electric_automobile', 'electric_car',
'gas_guzzler', 'hardtop', 'hatchback',
'horseless_carriage', 'hot_rod', 'hot-rod',
'jeep', 'landrover', 'limousine', 'limo',
'loaner', 'minicar', 'minivan', 'Model_T',
'pace_car', 'racer', 'race_car', 'racing_car',
'roadster', 'runabout', 'two-seater', 'sedan',
'saloon', 'sport_utility',
'sport_utility_vehicle', 'S.U.V.', 'SUV',
'sports_car', 'sport_car', 'Stanley_Steamer',
'stock_car', 'subcompact', 'subcompact_car',
'touring_car', 'phaeton', 'tourer', 'used-
car', 'secondhand_car']
```

66 hyponyms of car sense 1

not included



- *direct hyponym* / ***full hyponym***
  - S: (n) ambulance#1 (a vehicle that takes people to and from hospitals)
    - S: (n) funny wagon#1 (an ambulance used to transport patients to a mental hospital)
  - S: (n) beach wagon#1, station wagon#1, wagon#5, estate car#1, beach waggon#1, station waggon#1, waggon#2 (a car that has a long body and rear door with space behind rear seat)
    - S: (n) shooting brake#1 (another name for a station wagon)
  - S: (n) bus#4, jalopy#1, heap#3 (a car that is old and unreliable) *"the fenders had fallen off that old bus"*
  - S: (n) cab#3, hack#5, taxi#1, taxicab#1 (a car driven by a person whose job is to take passengers where they want to go in exchange for money)
    - S: (n) gypsy cab#1 (a taxicab that cruises for customers although it is licensed only to respond to calls)
    - S: (n) minicab#1 (a minicar used as a taxicab)
  - S: (n) compact#3, compact car#1 (a small and economical car)
  - S: (n) convertible#1 (a car that has top that can be folded or removed)
  - S: (n) coupe#1 (a car with two doors and front seats and a luggage compartment)
  - S: (n) cruiser#1, police cruiser#1, patrol car#1, police car#1, prowl car#1, squad car#1 (a car in which policemen cruise the streets; equipped with radiotelephonic communications to headquarters)
    - S: (n) panda car#1 (a police cruiser)
  - S: (n) electric#1, electric automobile#1, electric car#1 (a car that is powered by electricity)
  - S: (n) gas guzzler#1 (a car with relatively low fuel efficiency)
  - S: (n) hardtop#1 (a car that resembles a convertible but has a fixed rigid top)
  - S: (n) hatchback#1 (a car having a hatchback door)
  - S: (n) horseless carriage#1 (an early term for an automobile) *"when*

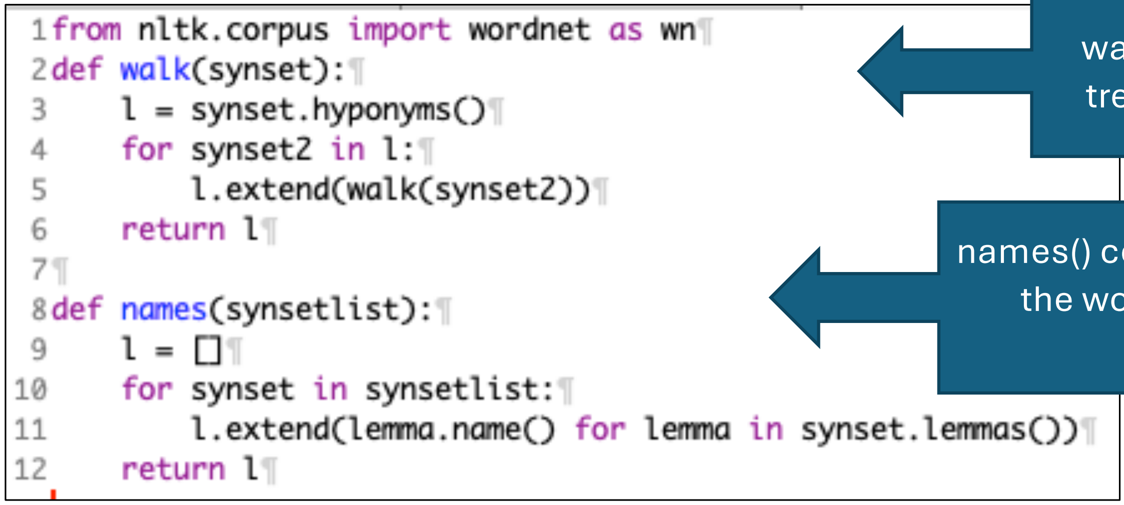
Full hyponymy

- Let's descend:

```
>>> s = wn.synset('car.n.1')
>>> s.hyponyms()[0]
Synset('ambulance.n.01')
>>> s.hyponyms()[0].hyponyms()
[Synset('funny_wagon.n.01')]
>>> s.hyponyms()[0].hyponyms()[0]
Synset('funny_wagon.n.01')
>>> s.hyponyms()[0].hyponyms()[0].hyponyms()
[]
```

Full hyponymy

`fullhyponyms.py`



```python
from nltk.corpus import wordnet as wn
def walk(synset):
    l = synset.hyponyms()
    for synset2 in l:
        l.extend(walk(synset2))
    return l

def names(synsetlist):
    l = []
    for synset in synsetlist:
        l.extend(lemma.name() for lemma in synset.lemmas())
    return l
```

walk() walks the hyponymy tree collecting the synsets

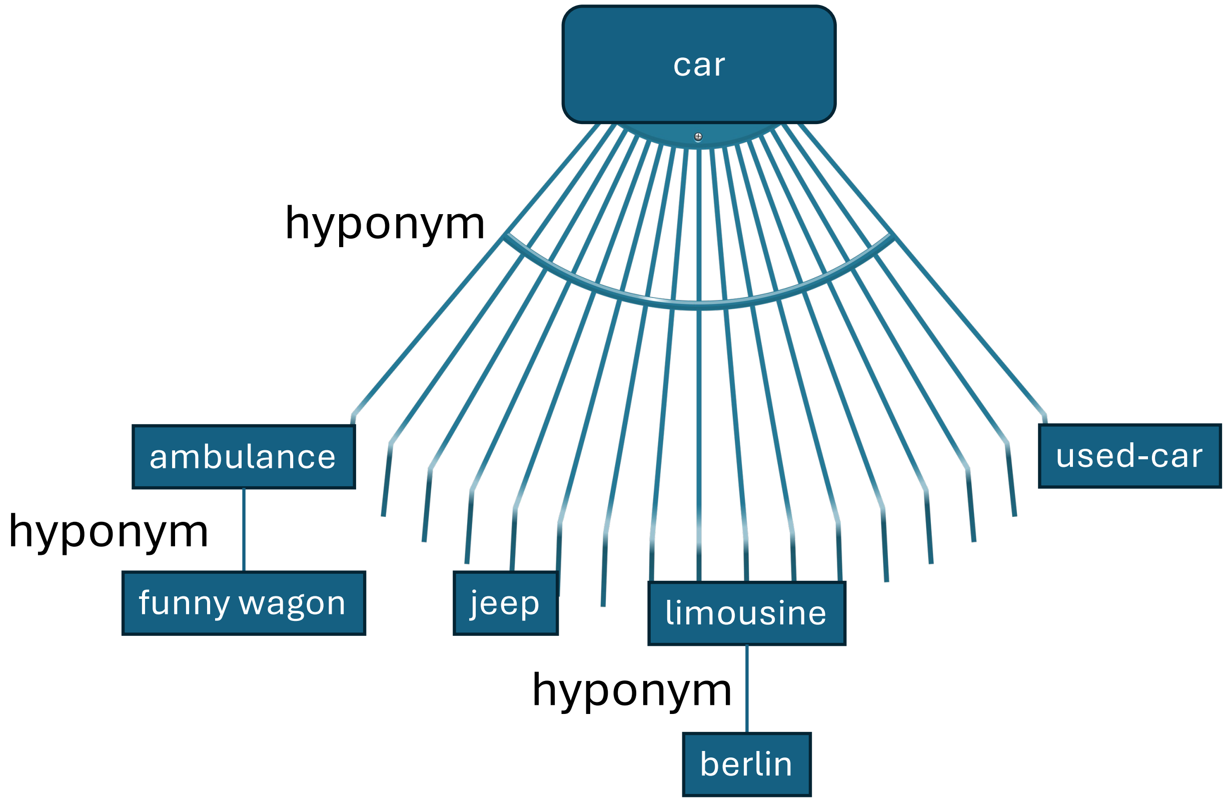
names() computes the lemma names, i.e. the words that belong to the list of synsets

```
$ python -i fullhyponyms.py
>>> s = wn.synset('car.n.1')
>>> names(walk(s))
```

Full hyponymy

```
>>> names(walk(s))
['ambulance', 'beach_wagon', 'station_wagon', 'wagon',
'estate_car', 'beach_waggon', 'station_waggon', 'waggon', 'bus',
'jalopy', 'heap', 'cab', 'hack', 'taxi', 'taxicab', 'compact',
'compact_car', 'convertible', 'coupe', 'cruiser', 'police_cruiser',
'patrol_car', 'police_car', 'prowl_car', 'squad_car', 'electric',
'electric_automobile', 'electric_car', 'gas_guzzler', 'hardtop',
'hatchback', 'horseless_carriage', 'hot_rod', 'hot-rod', 'jeep',
'landrover', 'limousine', 'limo', 'loaner', 'minicar', 'minivan',
'Model_T', 'pace_car', 'racer', 'race_car', 'racing_car',
'roadster', 'runabout', 'two-seater', 'sedan', 'saloon',
'sport_utility', 'sport_utility_vehicle', 'S.U.V.', 'SUV',
'sports_car', 'sport_car', 'Stanley_Steamer', 'stock_car',
'subcompact', 'subcompact_car', 'touring_car', 'phaeton', 'tourer',
'used-car', 'secondhand_car', 'funny_wagon', 'shooting_brake',
'gypsy_cab', 'minicab', 'panda_car', 'berlin', 'minicab',
'finisher', 'stock_car', 'brougham']
>>> len(names(walk(s)))
76
```

Full hyponymy

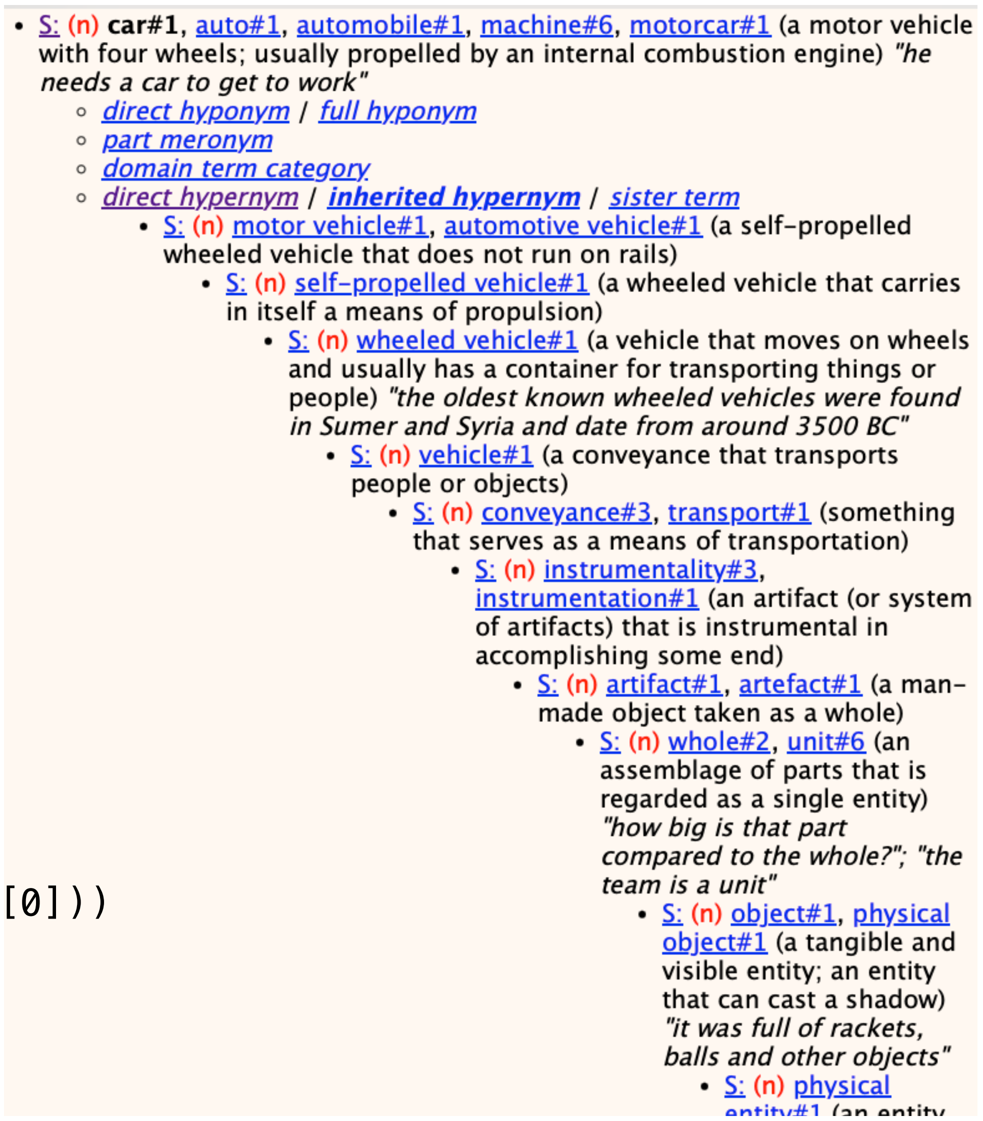


Full hyponymy

Let's run names(walk(*synset*)) on some hypernyms of car#1

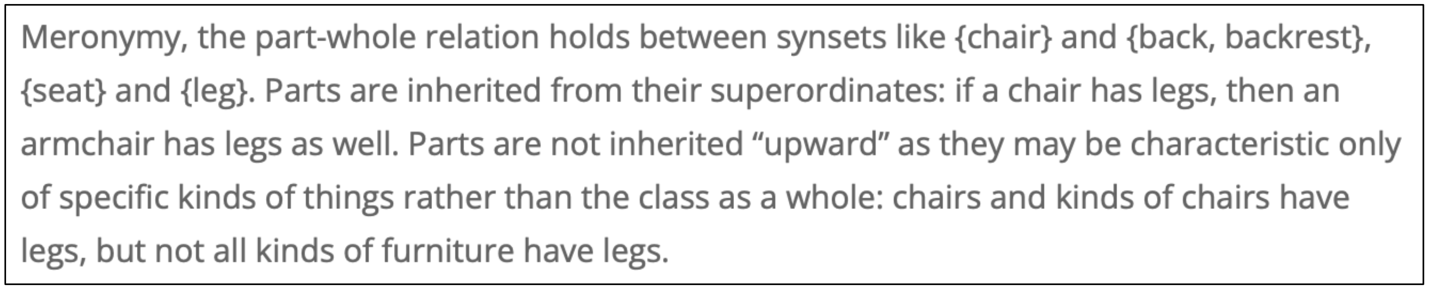
e.g.

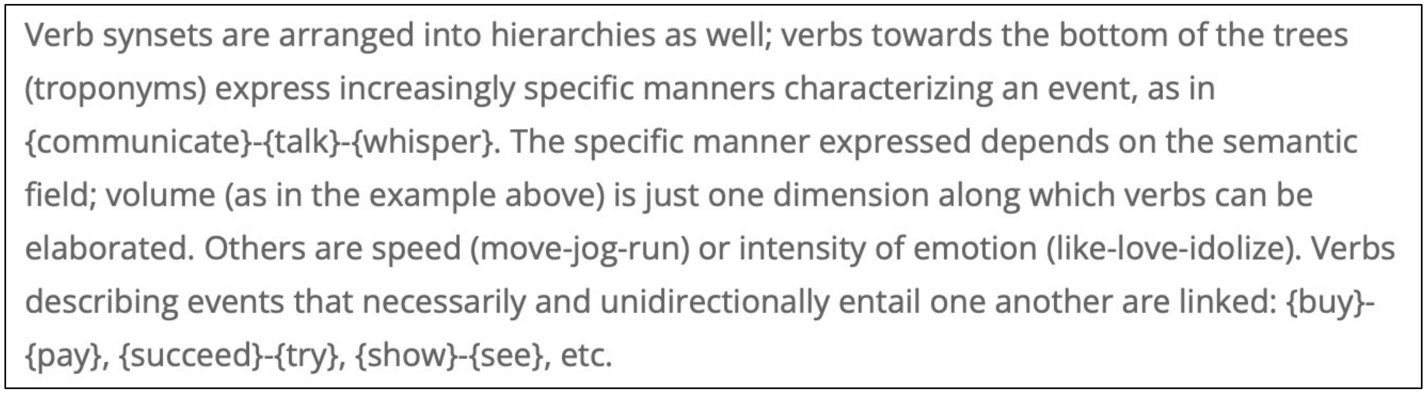
`names(walk(wn.synset('car.n.1').hypernyms()[0]))`



- S: (n) **car#1**, auto#1, automobile#1, machine#6, motorcar#1 (a motor vehicle with four wheels; usually propelled by an internal combustion engine) *"he needs a car to get to work"*
  - *direct hyponym* / *full hyponym*
  - *part meronym*
  - *domain term category*
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) motor vehicle#1, automotive vehicle#1 (a self-propelled wheeled vehicle that does not run on rails)
      - S: (n) self-propelled vehicle#1 (a wheeled vehicle that carries in itself a means of propulsion)
        - S: (n) wheeled vehicle#1 (a vehicle that moves on wheels and usually has a container for transporting things or people) *"the oldest known wheeled vehicles were found in Sumer and Syria and date from around 3500 BC"*
          - S: (n) vehicle#1 (a conveyance that transports people or objects)
            - S: (n) conveyance#3, transport#1 (something that serves as a means of transportation)
              - S: (n) instrumentality#3, instrumentation#1 (an artifact (or system of artifacts) that is instrumental in accomplishing some end)
                - S: (n) artifact#1, artefact#1 (a man-made object taken as a whole)
                  - S: (n) whole#2, unit#6 (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
                    - S: (n) object#1, physical object#1 (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
                      - S: (n) physical entity#1 (an entity

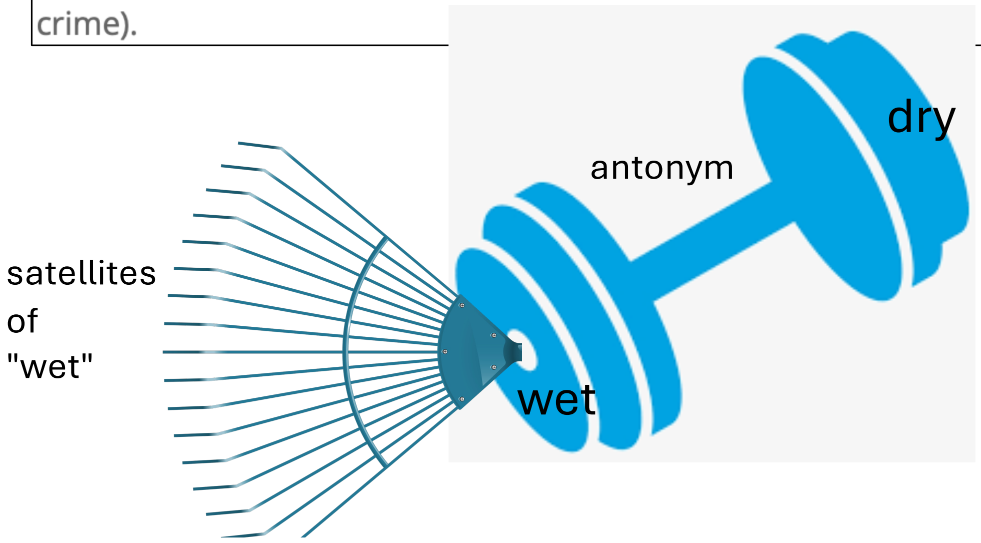
WordNet


Meronymy, the part-whole relation holds between synsets like {chair} and {back, backrest}, {seat} and {leg}. Parts are inherited from their superordinates: if a chair has legs, then an armchair has legs as well. Parts are not inherited “upward” as they may be characteristic only of specific kinds of things rather than the class as a whole: chairs and kinds of chairs have legs, but not all kinds of furniture have legs.

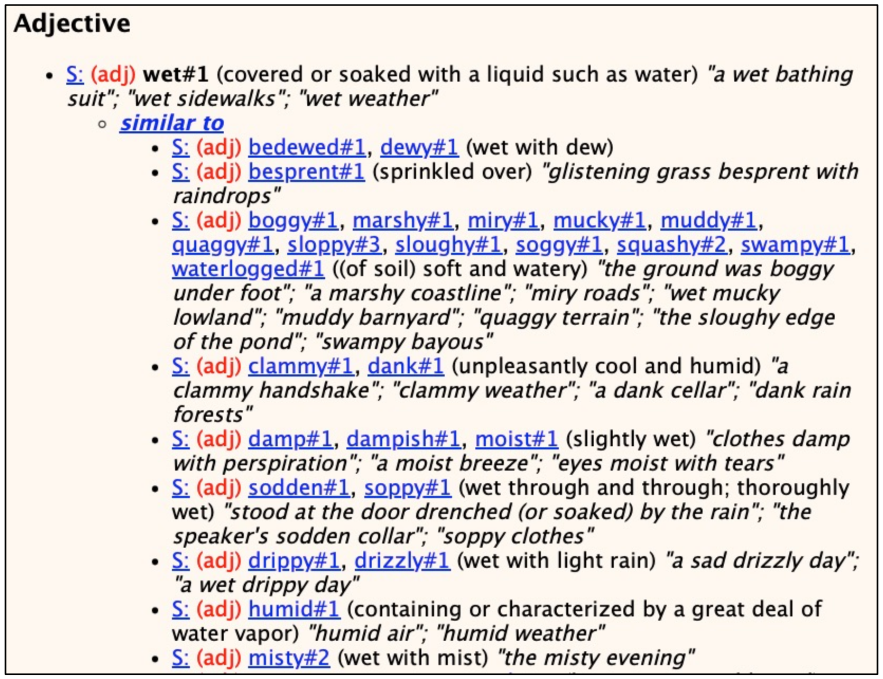

Verb synsets are arranged into hierarchies as well; verbs towards the bottom of the trees (troponyms) express increasingly specific manners characterizing an event, as in {communicate}-{talk}-{whisper}. The specific manner expressed depends on the semantic field; volume (as in the example above) is just one dimension along which verbs can be elaborated. Others are speed (move-jog-run) or intensity of emotion (like-love-idolize). Verbs describing events that necessarily and unidirectionally entail one another are linked: {buy}-{pay}, {succeed}-{try}, {show}-{see}, etc.

WordNet: adjectives and satellites

Adjectives are organized in terms of antonymy. Pairs of “direct” antonyms like wet-dry and young-old reflect the strong semantic contract of their members. Each of these polar adjectives in turn is linked to a number of “semantically similar” ones: dry is linked to parched, arid, dessicated and bone-dry and wet to soggy, waterlogged, etc. Semantically similar adjectives are “indirect antonyms” of the contral member of the opposite pole. Relational adjectives ("pertainyms") point to the nouns they are derived from (criminal-crime).

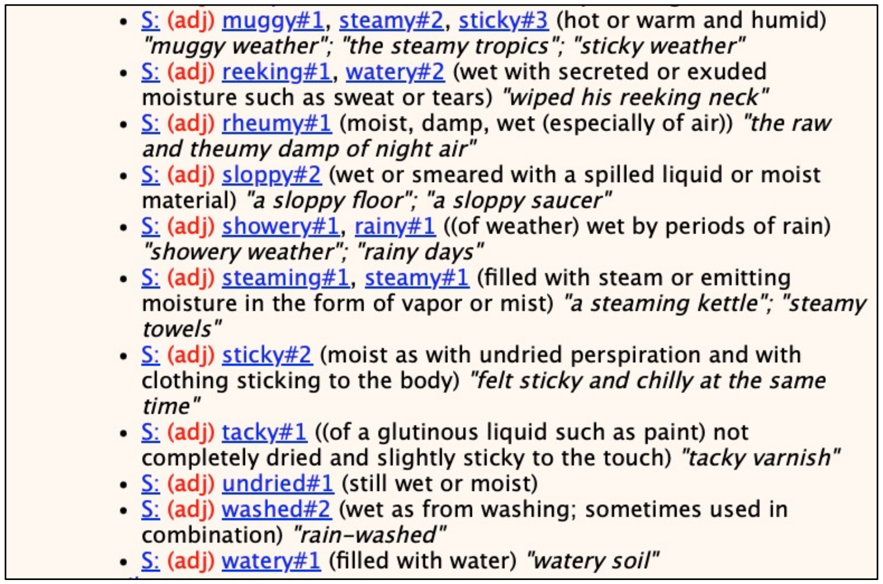


WordNet: satellites via `similar to`



Adjective

- S: (adj) **wet#1** (covered or soaked with a liquid such as water) *"a wet bathing suit"; "wet sidewalks"; "wet weather"*
    - ***similar to***
        - S: (adj) bedewed#1, dewy#1 (wet with dew)
        - S: (adj) besprent#1 (sprinkled over) *"glistening grass besprent with raindrops"*
        - S: (adj) boggy#1, marshy#1, miry#1, mucky#1, muddy#1, quaggy#1, sloppy#3, sloughy#1, soggy#1, squashy#2, swampy#1, waterlogged#1 ((of soil) soft and watery) *"the ground was boggy under foot"; "a marshy coastline"; "miry roads"; "wet mucky lowland"; "muddy barnyard"; "quaggy terrain"; "the sloughy edge of the pond"; "swampy bayous"*
        - S: (adj) clammy#1, dank#1 (unpleasantly cool and humid) *"a clammy handshake"; "clammy weather"; "a dank cellar"; "dank rain forests"*
        - S: (adj) damp#1, dampish#1, moist#1 (slightly wet) *"clothes damp with perspiration"; "a moist breeze"; "eyes moist with tears"*
        - S: (adj) sodden#1, soppy#1 (wet through and through; thoroughly wet) *"stood at the door drenched (or soaked) by the rain"; "the speaker's sodden collar"; "soppy clothes"*
        - S: (adj) drippy#1, drizzly#1 (wet with light rain) *"a sad drizzly day"; "a wet drippy day"*
        - S: (adj) humid#1 (containing or characterized by a great deal of water vapor) *"humid air"; "humid weather"*
        - S: (adj) misty#2 (wet with mist) *"the misty evening"*



- S: (adj) muggy#1, steamy#2, sticky#3 (hot or warm and humid) *"muggy weather"; "the steamy tropics"; "sticky weather"*
- S: (adj) reeking#1, watery#2 (wet with secreted or exuded moisture such as sweat or tears) *"wiped his reeking neck"*
- S: (adj) rheumy#1 (moist, damp, wet (especially of air)) *"the raw and theumy damp of night air"*
- S: (adj) sloppy#2 (wet or smeared with a spilled liquid or moist material) *"a sloppy floor"; "a sloppy saucer"*
- S: (adj) showery#1, rainy#1 ((of weather) wet by periods of rain) *"showery weather"; "rainy days"*
- S: (adj) steaming#1, steamy#1 (filled with steam or emitting moisture in the form of vapor or mist) *"a steaming kettle"; "steamy towels"*
- S: (adj) sticky#2 (moist as with undried perspiration and with clothing sticking to the body) *"felt sticky and chilly at the same time"*
- S: (adj) tacky#1 ((of a glutinous liquid such as paint) not completely dried and slightly sticky to the touch) *"tacky varnish"*
- S: (adj) undried#1 (still wet or moist)
- S: (adj) washed#2 (wet as from washing; sometimes used in combination) *"rain-washed"*
- S: (adj) watery#1 (filled with water) *"watery soil"*

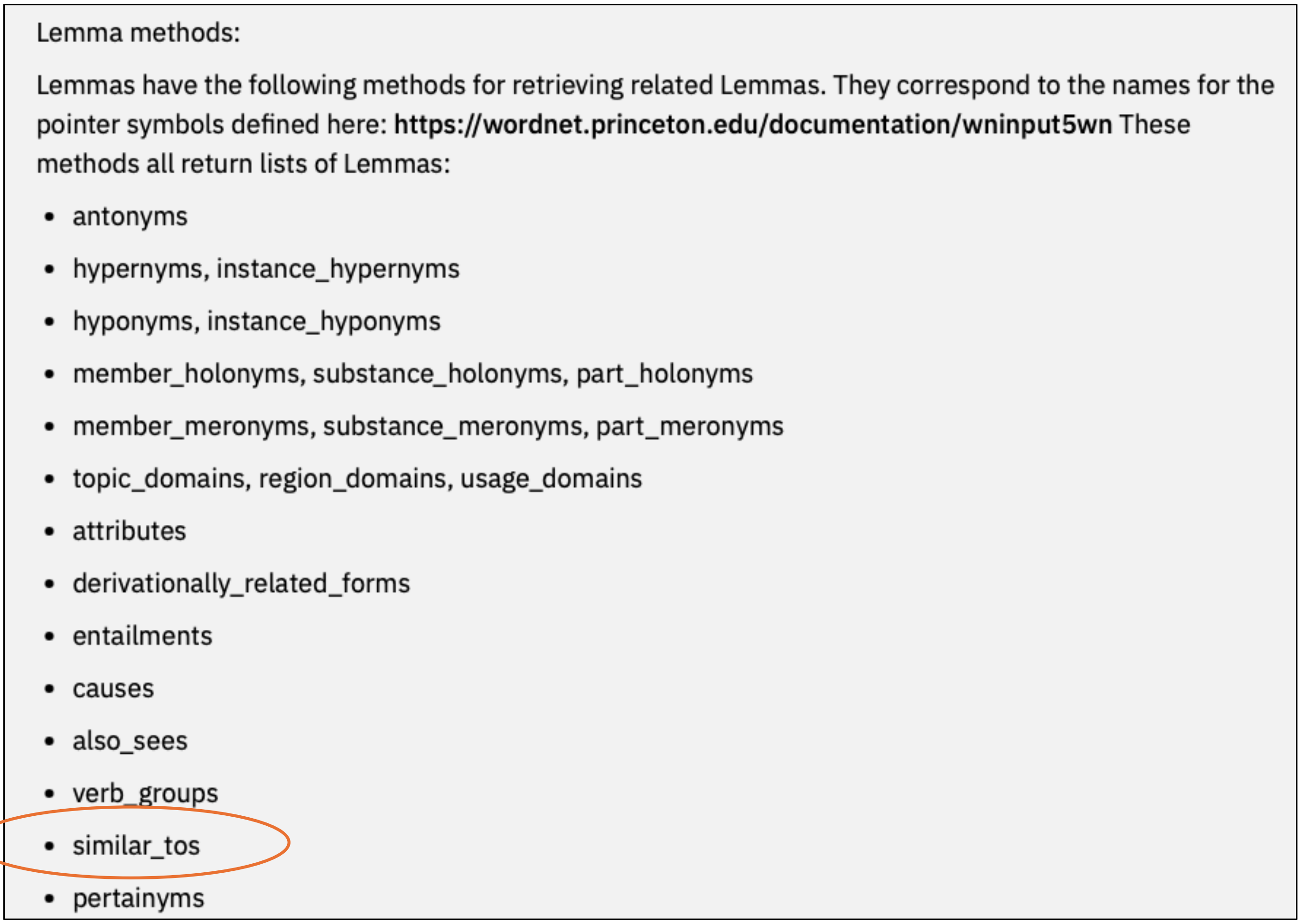
WordNet: satellites via `similar to`

- Antonyms are defined via **lemmas**, not **synsets**!

```
>>> wn.synset('wet.a.1')
Synset('wet.a.01')
>>> wn.synset('wet.a.1').antonyms()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'Synset' object has no attribute 'antonyms'
>>> wn.synset('wet.a.1').lemmas()[0].antonyms()
[Lemma('dry.a.01.dry')]
>>> wn.synset('wet.a.1').lemmas()[0].similar_tos()
[]
>>> wn.synset('wet.a.1').similar_tos()
[Synset('bedewed.s.01'), Synset('besprent.s.01'), Synset('boggy.s.01'),
Synset('clammy.s.01'), Synset('damp.s.01'), Synset('drippy.s.01'),
Synset('humid.s.01'), Synset('misty.s.02'), Synset('muggy.s.01'),
Synset('reeking.s.01'), Synset('rheumy.s.01'), Synset('showery.s.01'),
Synset('sloppy.s.02'), Synset('sodden.s.01'), Synset('steaming.s.01'),
Synset('sticky.s.02'), Synset('tacky.s.01'), Synset('undried.s.01'),
Synset('washed.s.02'), Synset('watery.s.01')]
```

Note: s for satellite!

Lemma relations

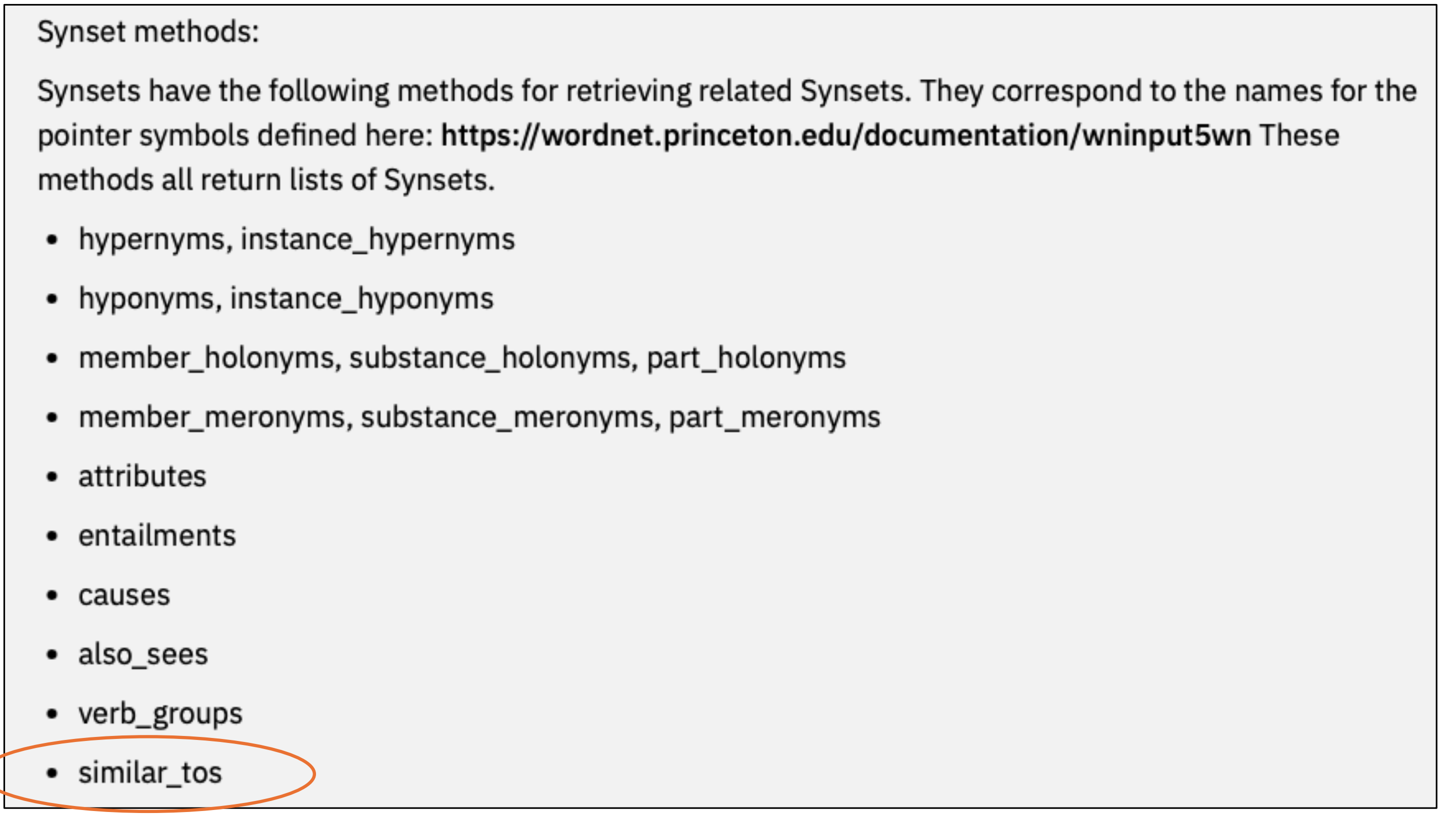


Lemma methods:

Lemmas have the following methods for retrieving related Lemmas. They correspond to the names for the pointer symbols defined here: **https://wordnet.princeton.edu/documentation/wninput5wn** These methods all return lists of Lemmas:

- antonyms
- hypernyms, instance_hypernyms
- hyponyms, instance_hyponyms
- member_holonyms, substance_holonyms, part_holonyms
- member_meronyms, substance_meronyms, part_meronyms
- topic_domains, region_domains, usage_domains
- attributes
- derivationally_related_forms
- entailments
- causes
- also_sees
- verb_groups
- similar_tos
- pertainyms

Synset relations



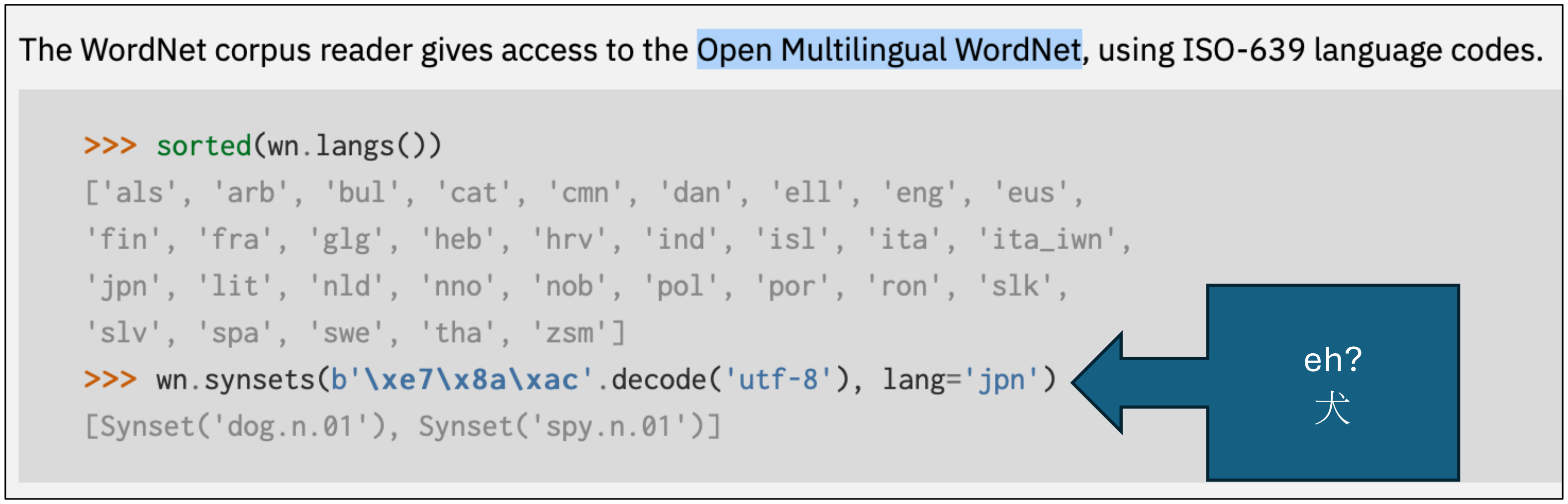
Synset methods:

Synsets have the following methods for retrieving related Synsets. They correspond to the names for the pointer symbols defined here: **https://wordnet.princeton.edu/documentation/wninput5wn** These methods all return lists of Synsets.

- hypernyms, instance_hypernyms
- hyponyms, instance_hyponyms
- member_holonyms, substance_holonyms, part_holonyms
- member_meronyms, substance_meronyms, part_meronyms
- attributes
- entailments
- causes
- also_sees
- verb_groups
- similar_tos

NLTK and WordNet

- Interlingua is English WordNet senses



The WordNet corpus reader gives access to the Open Multilingual WordNet, using ISO-639 language codes.

```
>>> sorted(wn.langs())
['als', 'arb', 'bul', 'cat', 'cmn', 'dan', 'ell', 'eng', 'eus',
'fin', 'fra', 'glg', 'heb', 'hrv', 'ind', 'isl', 'ita', 'ita_iwn',
'jpn', 'lit', 'nld', 'nno', 'nob', 'pol', 'por', 'ron', 'slk',
'slv', 'spa', 'swe', 'tha', 'zsm']
>>> wn.synsets(b'\xe7\x8a\xac'.decode('utf-8'), lang='jpn')
[Synset('dog.n.01'), Synset('spy.n.01')]
```

NLTK and WordNet

- Assuming your Terminal **and** Python installation accepts all Unicode characters ('utf-8'):

```
(venv) (base) ~$ python3
>>> import nltk
>>> from nltk.corpus import wordnet as wn
>>> wn.synsets('犬')
[]
>>> wn.synsets('犬', lang='jpn')
[Synset('dog.n.01'), Synset('spy.n.01')]
>>> wn.synsets('chien', lang='fra')
[Synset('dog.n.01'), Synset('pooch.n.01'), Synset('hound.n.01'),
Synset('andiron.n.01'), Synset('pawl.n.01'), Synset('frank.n.02'),
Synset('cad.n.01'), Synset('dog.n.03'), Synset('frump.n.01')]
>>> wn.synsets('cane', lang='ita')
[Synset('dog.n.01'), Synset('cramp.n.02'), Synset('hammer.n.01'),
Synset('bad_person.n.01'), Synset('incompetent.n.01')]
```

NLTK and WordNet

- macOS:
- **Disappointed**: my anaconda Python install didn't work for this.
  - SyntaxError: (unicode error) 'utf-8' codec can't decode bytes in position 2-3: invalid continuation byte
- Use Homebrew (https://brew.sh) Python instead with a virtual environment:
  1. install Homebrew first (*see above link*)
  2. brew install python3
     - `==> python@3.12`
     - Python has been installed as /opt/homebrew/bin/python3
  3. install nltk for Homebrew's Python (*use a virtual environment, let's call it venv*)
     - `$ /opt/homebrew/bin/python3 -m venv ~/venv`
     - `$ source ~/venv/bin/activate`
     - `(venv) (base) ~$ which python3`
     - `/Users/sandiway/venv/bin/python3`
     - `(venv) (base) ~$ python3 -m pip install nltk (venv) (base) ~$ python3`
     - `Python 3.12.3 (main, Apr  9 2024, 08:09:14) [Clang 15.0.0 (clang-1500.3.9.4)] on darwin`
     - `>>> import nltk`
     - `>>> from nltk.corpus import wordnet as wn`

NLTK and WordNet

using pip3 outside a virtual environment is not recommended!

```
$ /opt/homebrew/bin/pip3 install nltk
error: externally-managed-environment
× This environment is externally managed
╰─> To install Python packages system-wide, try brew install
    xyz, where xyz is the package you are trying to
    install.
    If you wish to install a Python library that isn't in Homebrew,
    use a virtual environment:
    python3 -m venv path/to/venv
    source path/to/venv/bin/activate
    python3 -m pip install xyz
```

NLTK and WordNet

- Turns out I should have just updated my anaconda Python:

```
$ conda update conda
$ conda update --all
$ conda update python
$ which python
/Users/sandiway/opt/anaconda3/bin/python
(base) ~$ python
Python 3.9.16 | packaged by conda-forge | (main, Feb  1 2023,
21:38:11)
>>> import nltk
>>> from nltk.corpus import wordnet as wn
>>> wn.synsets('犬', lang='jpn')
[Synset('dog.n.01'), Synset('spy.n.01')]
```