

# LING/C SC 581:

## Advanced Computational Linguistics

Lecture 16

# Announcements

---

There is no lecture 15

Lecture 14 was pre-recorded





京都・名古屋・東京  
for Kyōto, Nagoya, Tōkyō

列車名 Train No.	時刻 Time	行先 Destination	番線 Tracks	自由席	編成
96	14:30	東京 Tōkyō	27	1-3号車	16両
400	14:39	東京 Tōkyō	27	1-3号車	始発 16両
28	14:45	東京 Tōkyō	27	1-3号車	16両
402	14:51	東京 Tōkyō	24	1-3号車	始発 16両
のぞみ NOZOMI 230	15:00	東京 Tōkyō	24	1-3号車	始発 16両

「特大荷物スペースつき座席」を必ずご予約下さ





## 581 is an advanced class

- Any interest in my Keio lecture on the latest Chomsky proposals about the theory of language?

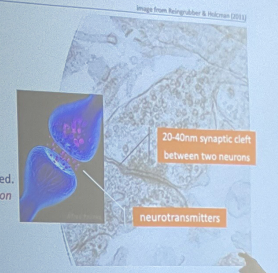
### Fact: Brain is slow

Computational efficiency (and bandwidth) are important considerations for all organic systems:

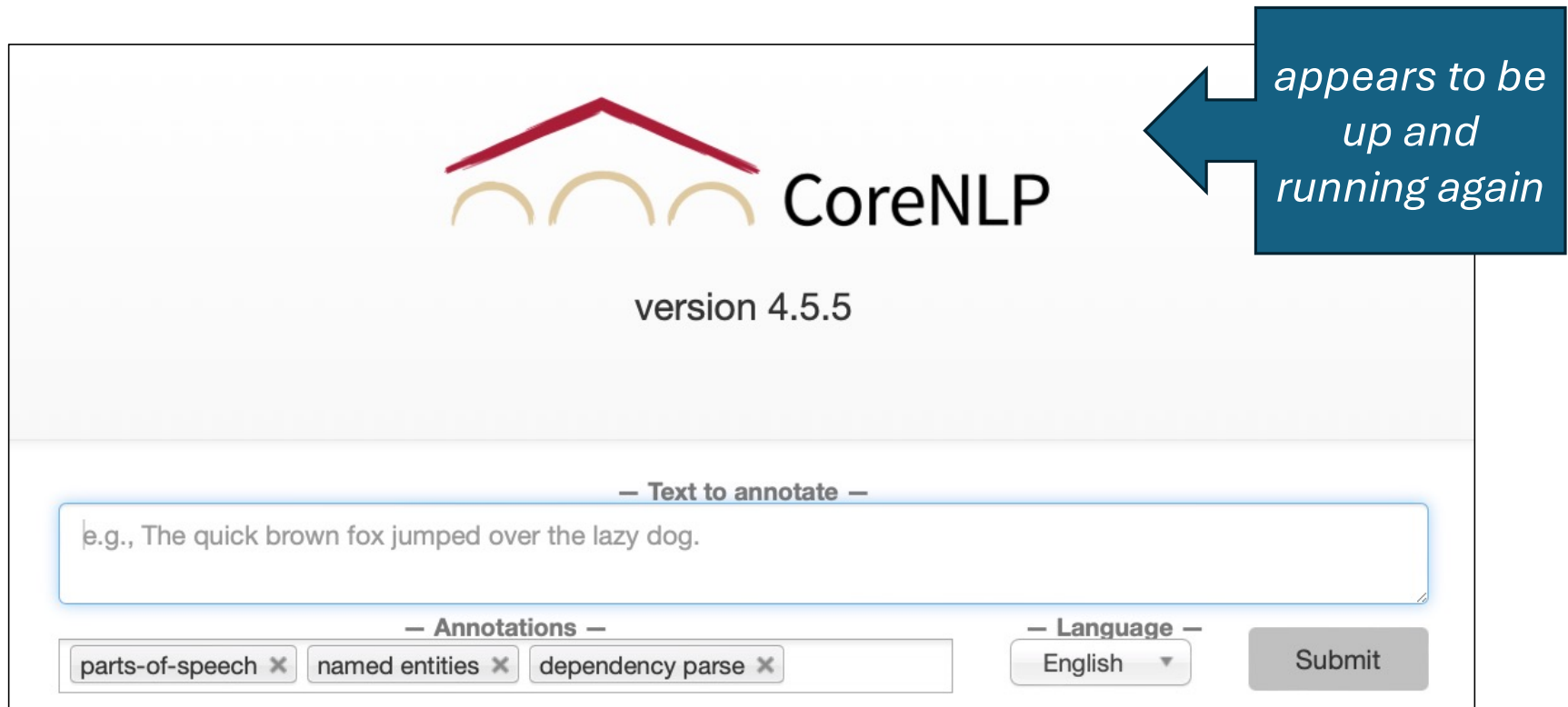
our sensory apparatus can generate vast amounts of data (*sensor mismatch*)

a slow (*chemical*) brain limits what can be analyzed

*The War of Soups and Sparks* (Valenstein, 2005) 19<sup>th</sup> century belief that neurons were electrically connected. Neurophysiologists believed only electrical transmission is fast enough to activate skeletal muscles. Mid-20<sup>th</sup> century: biochemical.



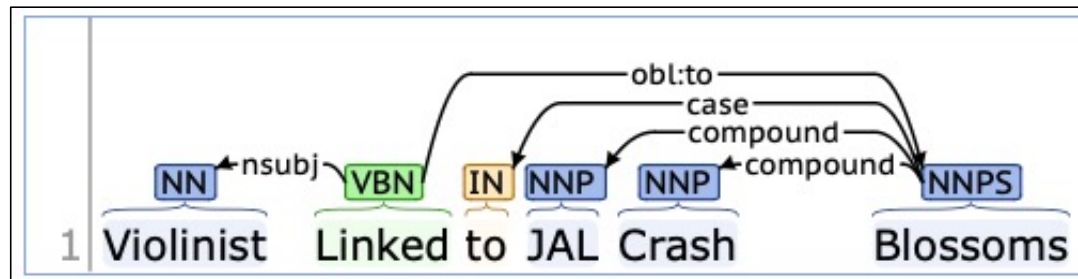
<https://corenlp.run>



The screenshot displays the CoreNLP web interface. At the top center is the CoreNLP logo, which consists of a red roof-like shape above three yellow arches, followed by the text "CoreNLP" and "version 4.5.5" below it. A blue callout box with a white arrow points to the logo area, containing the text "appears to be up and running again". Below the logo is a text input field with the placeholder text "e.g., The quick brown fox jumped over the lazy dog." and the label "— Text to annotate —". Underneath the input field is the "— Annotations —" section, which includes three buttons: "parts-of-speech x", "named entities x", and "dependency parse x". To the right of these buttons is a "— Language —" dropdown menu currently set to "English". A "Submit" button is located to the right of the language dropdown.

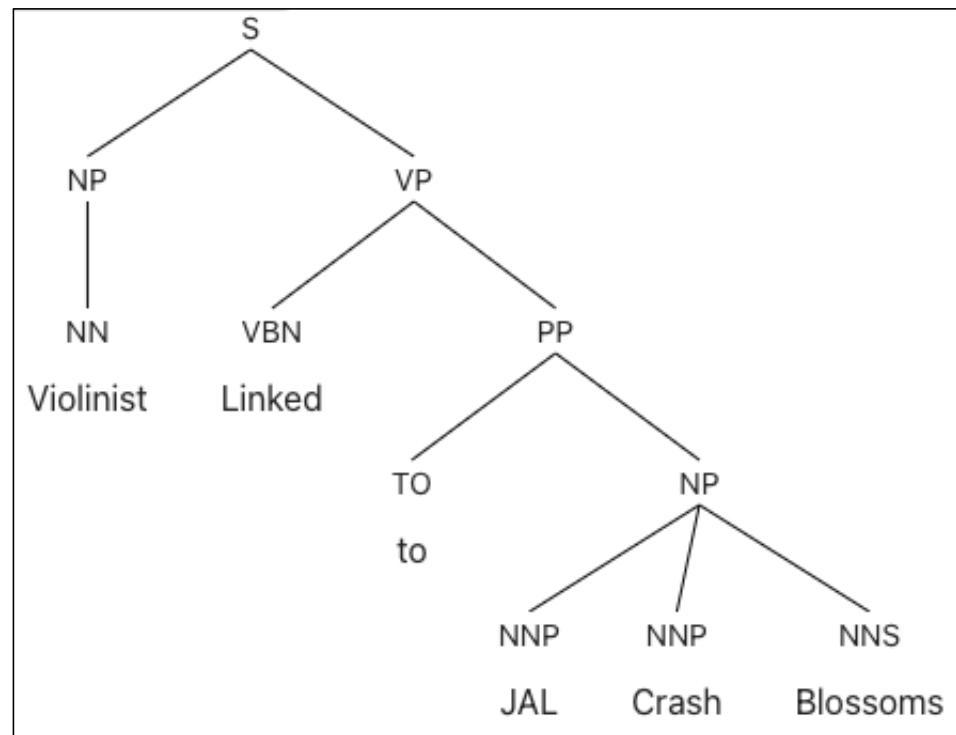
# *Violinist Linked to JAL Crash Blossoms*

- Crash Blossom Homework:
  - Not yet graded
- *Violinist Linked to JAL Crash Blossoms*



# *Violinist Linked to JAL Crash Blossoms*

- <https://parser.kitaev.io>
- Berkeley Neural Parser





# *Violinist Linked to JAL Crash Blossoms*

- Stanford Stanza Parser

```
$ python
Python 3.9.16 | packaged by conda-forge | (main, Feb 1 2023, 21:38:11)
[Clang 14.0.6 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import stanza
>>> nlp = stanza.Pipeline('en')
2024-03-14 06:19:20 INFO: Checking for updates to resources.json in case models have been
updated. Note: this behavior can be turned off with download_method=None or
download_method=DownloadMethod.REUSE_RESOURCES
2024-03-14 06:19:41 INFO: Using device: cpu
2024-03-14 06:19:41 INFO: Loading: tokenize
2024-03-14 06:19:41 INFO: Loading: pos
2024-03-14 06:19:41 INFO: Loading: lemma
2024-03-14 06:19:41 INFO: Loading: constituency
2024-03-14 06:19:41 INFO: Loading: depparse
2024-03-14 06:19:41 INFO: Loading: sentiment
2024-03-14 06:19:41 INFO: Loading: ner
2024-03-14 06:19:42 INFO: Done loading processors!
```

# Stanza updating

Downloading [https://raw.githubusercontent.com/stanfordnlp/stanza-resources/main/resources\\_1.5.0.json](https://raw.githubusercontent.com/stanfordnlp/stanza-resources/main/resources_1.5.0.json): 216kB [00:00, 67.0MB/s]

Downloading <https://huggingface.co/stanfordnlp/stanza-en/resolve/v1.5.0/models/>  
2024-03-14 06:19:41 INFO: Loading these models for language: en (English):

```
=====
```

Processor	Package
tokenize	combined
pos	combined
lemma	combined
constituency	wsj
depparse	combined
sentiment	sstplus
ner	ontonotes

```
=====
```

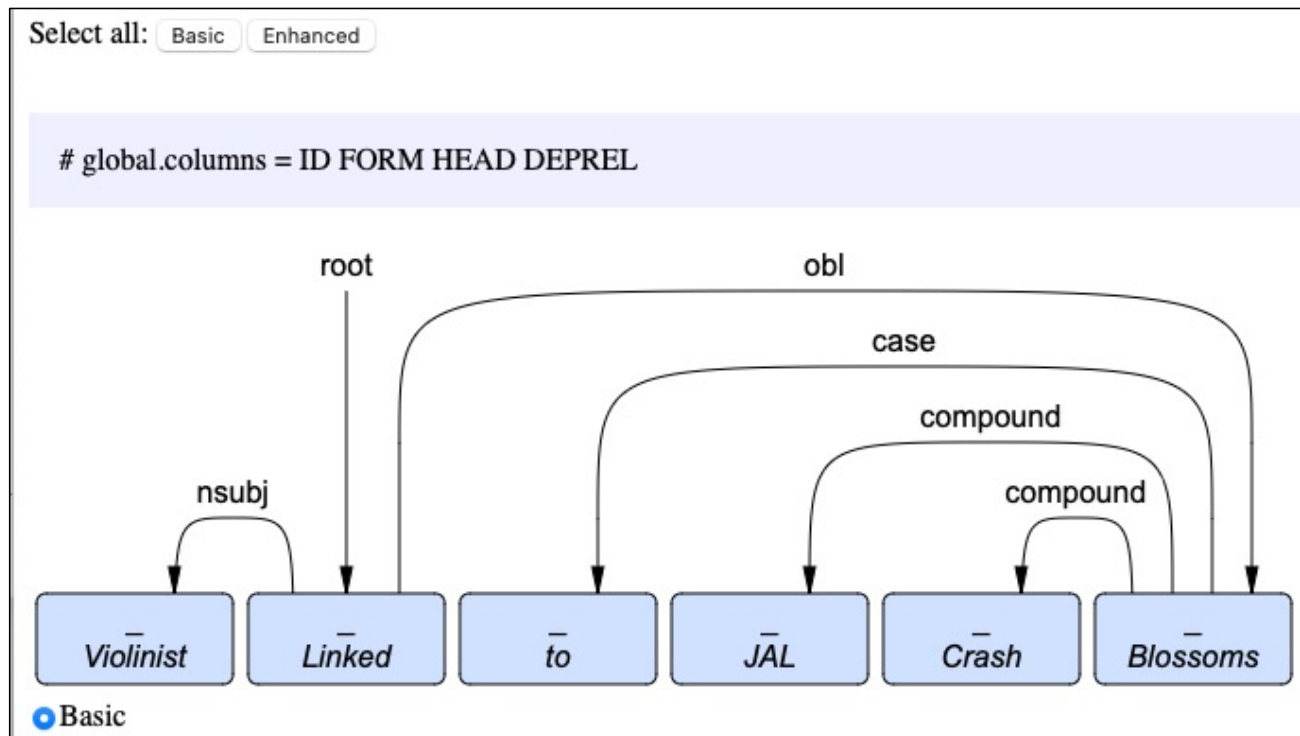
# *Violinist Linked to JAL Crash Blossoms*

```
>>> doc = nlp("Violinist Linked to JAL Crash Blossoms")
>>> s = ''
>>> words = doc.sentences[0].words
>>> for i,w in enumerate(words):
    s += '{:<3d}\t{:12s}\t{:<3d}\t{:15s}\n'.format(i+1,w.text,w.head,w.deprel)
...
>>> print(s)
1  Violinist      2  nsubj
2  Linked         0  root
3  to             6  case
4  JAL            6  compound
5  Crash          6  compound
6  Blossoms      2  obl
```

```
# global.columns = ID FORM HEAD DEPREL
```

# Violinist Linked to JAL Crash Blossoms

<https://urd2.let.rug.nl/~kleiweg/conllu/>



# Violinist Linked to JAL Crash Blossoms

<https://urd2.let.rug.nl/~kleiweg/conllu/>

Upload a file with one or more sentences annotated in [CoNLL-U](#) format:

Choose File no file selected

Submit

Here is an [example](#)

— — *OR* — —

Enter something in CoNLL-U format here:

```
# global.columns = ID FORM HEAD DEPREL
1 Violinist 2 nsubj
2 Linked 0 root
3 to 6 case
4 JAL 6 compound
5 Crash 6 compound
6 Blossoms 2 obl
```

# Newspapers could check their headlines?



The image is a screenshot of a news article from the Daily News. At the top, there is a search icon and a menu icon on the left, the "DAILY NEWS | NEWS" logo in the center, and a three-dot menu icon on the right. Below the logo is a navigation bar with "Crime", "U.S.", "World", and "Politics" links. The main headline reads "KING: N.C. police kill unarmed deaf man using sign language". Below the headline is the author's profile picture, name "SHAUN KING", a "FOLLOW" button, and the text "NEW YORK DAILY NEWS Monday, August 22, 2016, 10:49 AM". At the bottom are social media sharing icons for Facebook, Twitter, and Email.

Crime U.S. World Politics

## KING: N.C. police kill unarmed deaf man using sign language

 SHAUN KING [FOLLOW](#)  
NEW YORK DAILY NEWS Monday, August 22, 2016, 10:49 AM

[f](#) [t](#) [✉](#)

# Was fixed!

- I went to the Daily News website:

## **Shaun King column from 2016: North Carolina police kill unarmed deaf man who was using sign language**

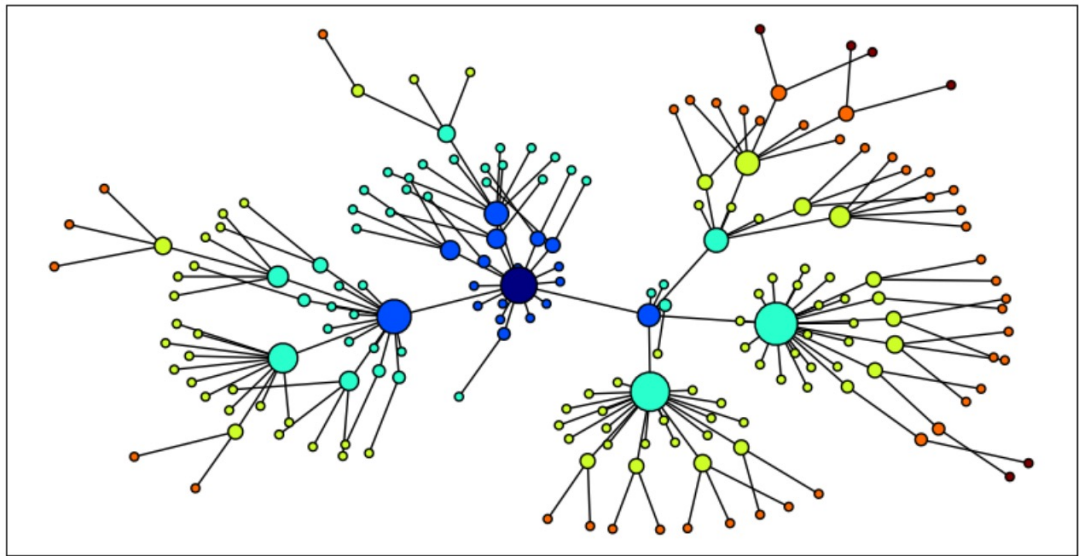
Aug 22, 2016 at 10:49 am

This is as bad as it gets.

A North Carolina state trooper shot and killed 29-year-old Daniel Harris — who was not only unarmed, but deaf — just feet from his home, over a speeding violation.

[According to early reports from neighbors](#) who witnessed the shooting this past Thursday night, Harris was shot and killed "almost immediately" after exiting his vehicle.

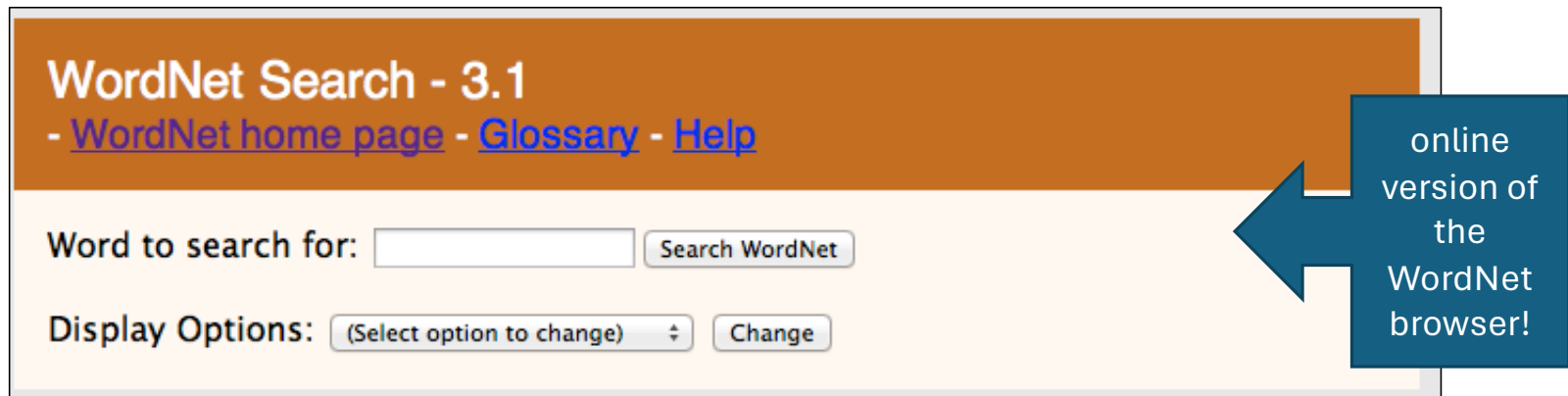
He appeared to be trying to communicate with the officer via sign language.





# WordNet: online interface

- WordNet 3.0
  - (3.1 the latest version but only online or the database files only)
  - <http://wordnetweb.princeton.edu/perl/webwn>



# WordNet: online interface

## 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>

## 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"*

WordNet Search - 3.1  
- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:  Search WordNet

Display Options:  (Select option to change)

- Hide Example Sentences
- Hide Glosses
- Show Frequency Counts
- Show Database Locations
- Show Lexical File Info
- Show Lexical File Numbers
- Show Sense Keys
- Show Sense Numbers**
- Show all
- Hide all

turn this on!

# 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

# WordNet 3.1 Demo

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 3.1 Demo

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).



# 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
```

## Words

Look up a word using `synsets()`; this function has an optional `pos` argument which lets you constrain the part of speech of the word:

```
>>> wn.synsets('dog')
[Synset('dog.n.01'), Synset('frump.n.01'), Synset('dog.n.03'), Synset('cad.n.01'),
Synset('frank.n.02'), Synset('paw1.n.01'), Synset('andiron.n.01'), Synset('chase.v.01')]
>>> wn.synsets('dog', pos=wn.VERB)
[Synset('chase.v.01')]
```

The other parts of speech are `NOUN`, `ADJ` and `ADV`. A synset is identified with a 3-part name of the form: `word.pos.nn`:

# nltk WordNet Notation

Details here:

- <http://www.nltk.org/howto/wordnet.html>
- A synset is uniquely identified with a 3-part name of the form: `word.pos.nn`
  - "head" of the synset is the first listed name: `word`
  - `pos`: one of [asrnv] (adjective/satellite/adverb/noun/verb)
- A lemma is uniquely identified with a 4-part name : `word.pos.nn.name`
  - the 3-part prefix is the synset

# nltk WordNet Notation

- Examples:

```
>>> 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')]
```

```
>>> wn.synset('motorbike.n.1')
Synset('minibike.n.01')
>>> wn.synset('motorbike.n.1').lemmas()
[Lemma('minibike.n.01.minibike'),
Lemma('minibike.n.01.motorbike')]
```




# NLTK and WordNet

Test your nltk:

```
>>> from nltk.corpus import wordnet as wn
>>> wn.synsets('cat')
[Synset('cat.n.01'),
Synset('guy.n.01'),
Synset('cat.n.03'),
Synset('kat.n.01'),
Synset('cat-o'-nine-tails.n.01'),
Synset('caterpillar.n.02'),
Synset('big_cat.n.01'),
Synset('computerized_tomography.n.01'),
Synset('cat.v.01'),
Synset('vomit.v.01')]
>>> s = wn.synsets('cat')
>>> s[6]
Synset('big_cat.n.01')
```

```
>>> s[6].lemma_names()
['big_cat', 'cat']
>>> s[6].lemma_names('fra')
['chat', 'fauve', 'félin']
>>> s[6].lemma_names('spa')
[]
>>> s.lemma_names('jpn')
['大型ネコ科動物']
>>> s[6].hypernyms()
[Synset('feline.n.01')]
>>> s[6].hypernyms()[0].hypernyms()
[Synset('carnivore.n.01')]
>>> s[6].hypernyms()[0].hypernyms()[0].hypernyms()
[Synset('placental.n.01')]
>>> s[6].hypernyms()[0].hypernyms()[0].hypernyms()[0].hypernyms()
[Synset('mammal.n.01')]
```



Open  
Multilingual  
WordNet

# NLTK and WordNet

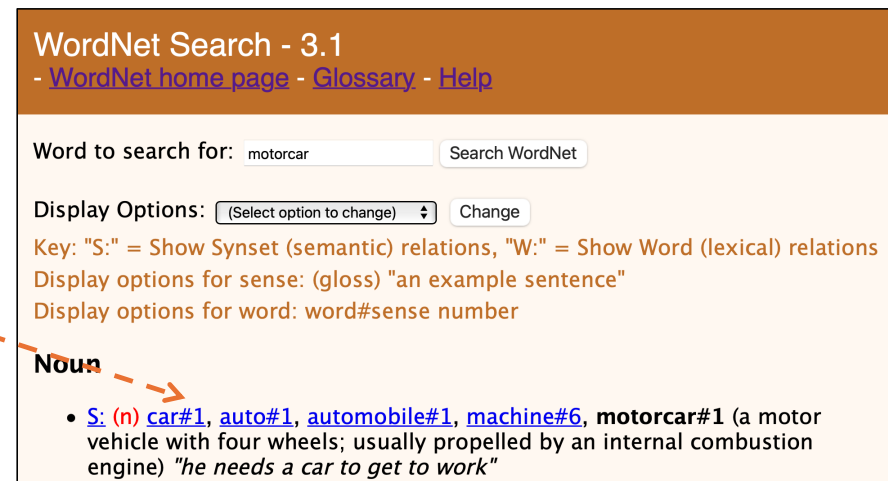
- Interlingua is English WordNet senses

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 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.01')
>>> 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:

Display Options:

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"*