WordNet Structure: Words, Senses, Synsets, and Relations
Manipulating wordnets with wn

Francis Bond

April 12, 2026

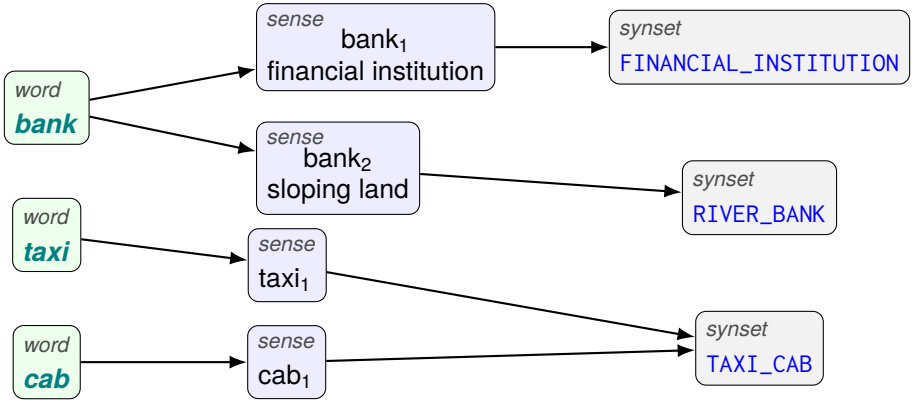
Faculty of Arts

Big idea: a lexicon organized by concepts

- A **word form** can express **multiple meanings** (polysemy).
- A **meaning** can be expressed by **multiple word forms** (synonymy).
- WordNets model meaning with:
  - **Words** (forms)
  - **Senses** (a word-in-a-particular-meaning)
  - **Synsets** (concepts; sets of near-synonymous senses)
  - **Relations** between synsets/senses

Faculty of Arts

Words → Senses → Synsets



Key: **senses** link **words** to **synsets** (concepts).

Faculty of Arts

What **lexicons** have

- A lexicon is a collection of synsets, senses, words and relations for a given language
- It will normally have some metadata associated with it
  - Label
  - License
  - Language
- The `wn` module has a collection of wordnets you can download easily
- If you have a wordnet in the LMF format, you can load it locally

Faculty of Arts

Loading and describing a lexicon in WN

```
>>> import wn
>>> wn.download('omw-en:1.4')
>>> ewn = wn.Wordnet('omw-en:1.4')
>>> print(ewn.describe())
Primary lexicons:
  omw-en:1.4
    Label  : OMW English Wordnet based on WordNet 3.0
    URL    : None
    License: https://wordnet.princeton.edu/license-and-commercial-use
    Words  : 156584 (a: 21538, n: 119034, r: 4481, v: 11531)
    Senses : 206978
    Synsets: 117659 (a: 7463, n: 82115, r: 3621, s: 10693, v: 13767)
    ILIs   : 117659
>>> wn.projects() # gives a list of downloadable wordnets
...
```

Adding, removing and listing lexicons with WN

```
>>> wn.add('path/to/omw-nb:1.4.xz')
>>> wn.remove('omw-nb:1.4')
>>> for lex in wn.lexicons():
...     print(f'{lex.id}:{lex.version}\t{lex.label}')
...
omw-en:1.4      OMW English Wordnet based on WordNet 3.0
omw-nb:1.4      Norwegian Wordnet (Bokmål)
odenet:1.3      Offenes Deutsches WordNet
ewn:2020        English WordNet
...
```

Faculty of Arts

What **Words** have

- **Canonical Form (Lemma)**: the written form(s) used in the lexicon
- **Part of speech**
- **Variant forms**
  - spelling variants (*colour/color*), scripts (犬 / いぬ), casing/diacritics policy, irregular forms (*child/children* [pl]),
- **Form metadata** (when present in a resource)
  - pronunciations, transliterations, orthographic variants type (US/UK)
  - morphological class (plural, past), if the resource encodes them at word-level
- **Derivations:** links to words related by derivation
- **Sense(s)**:
  - one `Word` can have multiple `Sense`s (polysemy)
  - all those senses share the same *string forms* but differ in meaning

Faculty of Arts

Form metadata example

```
<LexicalEntry id="ex-gato-n">
    <Lemma writtenForm="gato" partOfSpeech="n">
      <Tag category="gender">masculine</Tag>
    </Lemma>
    <Form writtenForm="gatos">
        <Tag category="number">plural</Tag>
    </Form>
</LexicalEntry>
```

- Grammatical properties such as gender, number, case, and tense can be represented with the ⟨Tag⟩ element.
- The ⟨Tag⟩ element has a category attribute which indicates the type of grammatical property, and the value is the text of the tag
- A universal tagging scheme is not prescribed — wordnet project owners are free to choose a scheme

Finding Words in WN

```
>>> wn.words('pencil')
[Word('ewn-pencil-n'), Word('ewn-pencil-v')]
>>> wn.words('pencil', pos='v')
[Word('ewn-pencil-v')]
len(wn.words())
311711
>>> len(wn.words(pos='v'))
29419
>>> len(wn.words(pos='v', lexicon='ewn'))
11595
>>> wn.word('ewn-pencil-n', lexicon='ewn')
Word('ewn-pencil-n')
```

Faculty of Arts

Exploring Words in WN

```
>>> w = wn.words('goose')[0]
>>> w.pos  # part of speech
'n'
>>> w.forms()  # other word forms (e.g., irregular inflections)
['goose', 'geese']
>>> w.lemma()  # canonical form
'goose'
>>> w.derived_words()
[Word('ewn-gosling-n'), Word('ewn-goosy-s'), Word('ewn-goosey-s')]
>>> w.senses()
[Sense('ewn-goose-n-01858313-01'), Sense('ewn-goose-n-10177319-06'), Sense('ewn-g
>>> w.synsets()
[Synset('ewn-01858313-n'), Synset('ewn-10177319-n'), Synset('ewn-07662430-n')]
>>> w.forms()[0].pronunciations()[0].value  # in oewn
"'gu:s"
```

What **Senses** have

- A **sense** is the pairing: **Word** $\times$ **Synset**
- Sense-level properties often include:
  - **Sense IDs / keys** (resource-specific identifiers)
  - **Frequency / counts** (e.g., corpus-based tagging counts)
  - **Derivations** (constrained by meaning)
  - **Usage examples** (example sentences tied to this meaning)
  - **Syntactic/verb frames** (common in some WordNet-style resources)
  - **Sense-level relations** such as antonymy
- Why sense-level? Because **frequency and examples depend on the meaning**, not just the string or the concept.

Faculty of Arts

Finding Senses in WN

```
>>> ewn.senses('plow', pos='n')
[Sense('ewn-plow-n-03973894-01')]
>>> ewn.sense('ewn-plow-v-01745745-01')
Sense('ewn-plow-v-01745745-01')
>>> len(ewn.senses(pos='n'))
146347
```

Faculty of Arts

Exploring Senses in WN

```
>>> s = ewn.senses('dark', pos='n')[0]
>>> s.word()    # each sense links to a single word
Word('ewn-dark-n')
>>> s.synset()  # each sense links to a single synset
Synset('ewn-14007000-n')
>>> s.get_related('antonym')
[Sense('ewn-light-n-14006789-01')]
>>> s.get_related('derivation')
[Sense('ewn-dark-a-00273948-01')]
>>> s.metadata()  # omw-en:1.4
{'identifier': 'dark%1:26:00::'}
>>> s.counts()
[2]
>>> ewn.senses('darken', pos='v')[0].frames()
['Something ----s']
```

What **Synsets** have

- A **synset** represents a **concept** (language-internal, but linkable).
- Typical synset-level properties:
  - **Definition / gloss** (concept description)
  - **Usage examples** (example sentences)
  - **Part of speech** (noun/verb/adj/adv in many resources)
  - **Semantic fields / domains** (e.g., *lexicographer files*)
  - **Concept relations**: hypernym, meronym, cause, etc.
  - **ILI links** (interlingual index), when available
- Why synset-level? Because these describe the **concept itself**, independent of which word expresses it.

Faculty of Arts

Finding Synsets in WN

```
>>> wn.synsets('scepter')
[Synset('ewn-14467142-n'), Synset('ewn-07282278-n')]
>>> wn.synset('ewn-07282278-n').ili
'i74874'
>>> wn.synsets(ili='i74874')
[Synset('ewn-07282278-n'), Synset('wnja-07267573-n'), Synset('frawn-07267573-n')]
```

Faculty of Arts

Exploring Synsets in WN I

```
>>> ss = wn.synsets('hound', pos='n')[0]
>>> ss.senses()
[Sense('ewn-hound-n-02090203-01'), Sense('ewn-hound_dog-n-02090203-02')]
>>> ss.words()
[Word('ewn-hound-n'), Word('ewn-hound_dog-n')]
>>> ss.lemmas()
['hound', 'hound dog']
>>> ss.definition()
'any of several breeds of dog used for hunting typically having large drooping ea
```

Faculty of Arts

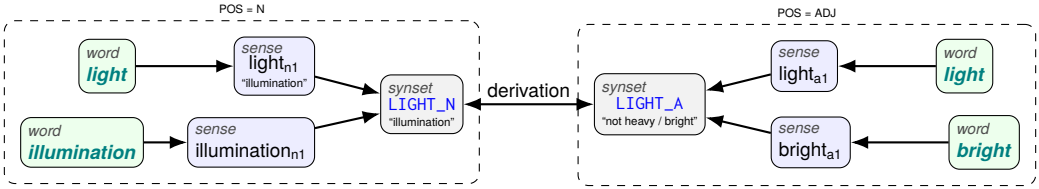
Exploring Synsets in WN II

```
>>> ss.hypernyms()
[Synset('ewn-02089774-n')]
>>> ss.hypernyms()[0].lemmas()
['hunting dog']
>>> len(ss.hyponyms())
20
>>> ss.hyponyms()[0].lemmas()
['Afghan', 'Afghan hound']
>>> ss.max_depth()
15
>>> ss.shortest_path(wn.synsets('dog', pos='n')[0])
[Synset('ewn-02090203-n'), Synset('ewn-02089774-n'), Synset('ewn-02086723-n')]
```

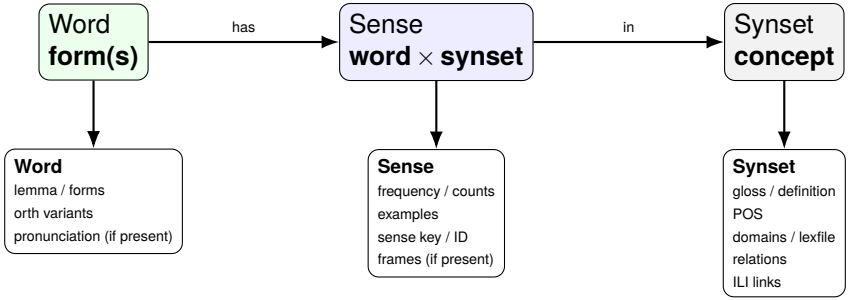

Faculty of Arts

POS-specific variants: concepts live in POS-specific synsets

- A synset is typically **POS-specific** (noun synset vs adjective synset, etc.).
- The “same general idea” across POS is modeled via **relations between synsets**, not by mixing POS inside one synset.



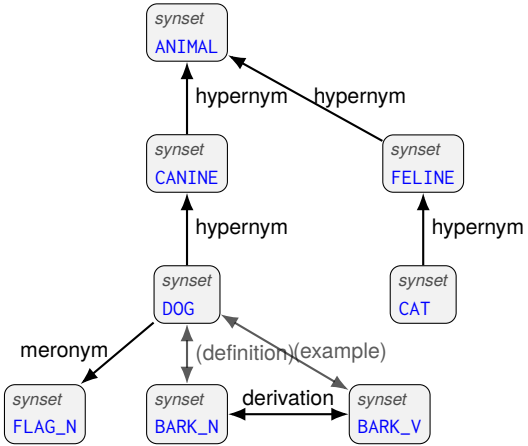
Cheat sheet: where does each kind of information live?



Rule of thumb: *form* → Word; *usage of a meaning* → Sense; *concept + relations* → Synset.

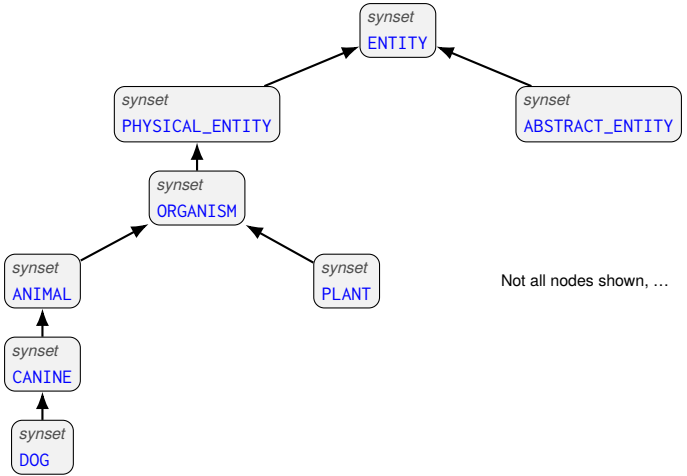
Faculty of Arts

Relations: building a graph over concepts



WordNets are **graphs**: taxonomy-like relations (e.g. hypernymy) plus many non-taxonomic links (meronymy, cause, related, used in example/definition etc.).

The taxonomy: a (mostly) rooted hierarchy for nouns



Multiple inheritance is possible, loops are not, for OEWN 2025, verbs are also rooted.

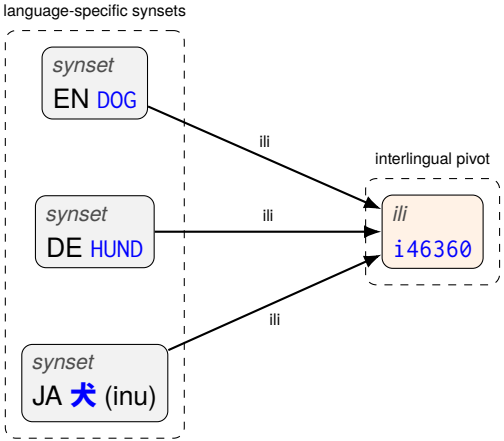
Lexicons, IDs, and versions (in wn)

- A **WordNet resource** is packaged as a **lexicon** (often: 1 lexicon per resource).
- wn supports multiple lexicons (and multiple versions) in one database.
- Lexicon specifiers (examples):
  - `ewn:2020` = a specific lexicon & version
  - `ewn:*` = all versions of ewn
  - `*` = all lexicons

Most of the time it is better to use a single lexicon (e.g. `ewn=wn.Wordnet('oewn:2024')`)

Faculty of Arts

Interlingual Index (ILI): linking concepts across languages



ILI enables “same concept, different languages” queries when lexicons provide ILI links.

Using the ILI

- Wordnets linked to the ILI can be connected
- Concepts with the same ILI should be the same across all languages
- So we would expect them to be translation equivalents
- If one language has a synset with an ILI and another doesn't then this should indicate a lexical gap.
  - Although sometimes it just means that the concept has not been added yet
- A synset can be marked as `Lexicalized=False` (default `True`)
  - This means that it has no native word
  - If it has senses, then they are compositional phrases (like 雌虎 *mesu.tora* "female tiger" for *tigress*)
  - Very few wordnets have this implemented (they were introduced by the Multiwordnet project)

Faculty of Arts

Translating using the ILI

```
# translating  a word gives a list of senses
>>> for sense, ja_words in w.translate(lang='ja').items():
...     print(sense, ja_words)
...
Sense('ewn-goose-n-01858313-01') [Word('wnja-n-1254'), Word('wnja-n-33090'), Word
Sense('ewn-goose-n-10177319-06') []
Sense('ewn-goose-n-07662430-01') [Word('wnja-n-1254')]
# Translating a sense gives a more focused list of senses
>>> s.translate(lang='fr')
[Sense('frawn-lex52992--13983515-n')]
>>> s.translate(lang='fr')[0].word().lemma()
'obscurité'
# Translating a synset gives a list of synsets (but normally just one)
>>> ss.translate(lang='fr')  # translation returns a list of synsets
[Synset('frawn-02087551-n')]
```

Some other useful functionality

- You can set the directory where `wn`'s data is stored.

```
>>> wn.config.data_directory = ".cache/wn"
```

  This is useful to keep the db small and focused
- You can use `wn` to validate a wordnet

```
$ python -m wn validate english-wordnet-2021.xml --output-file errors.json
```

- There is a whole command line interface where you can list projects (downloadable), lexicons (downloaded), load lexicons and more

Faculty of Arts

Summary: what to remember

1. **Synset** $\approx$ **concept node**; a set of synonyms, that makes up a concept. Can be non-lexicalized!
2. **Relations** turn synsets and senses into a richly typed graph.
3. **Taxonomy** (hypernymy) gives a hierarchy (esp. nouns).
4. **Lexicons** + versions allow reproducibility.
5. **ILI** enables cross-lingual alignment when available.
6. **Properties live at different layers**: form (Word), usage (Sense), concept (Synset).

Different Wordnet Projects contain different information, OMW tries to harmonize when possible

Faculty of Arts