



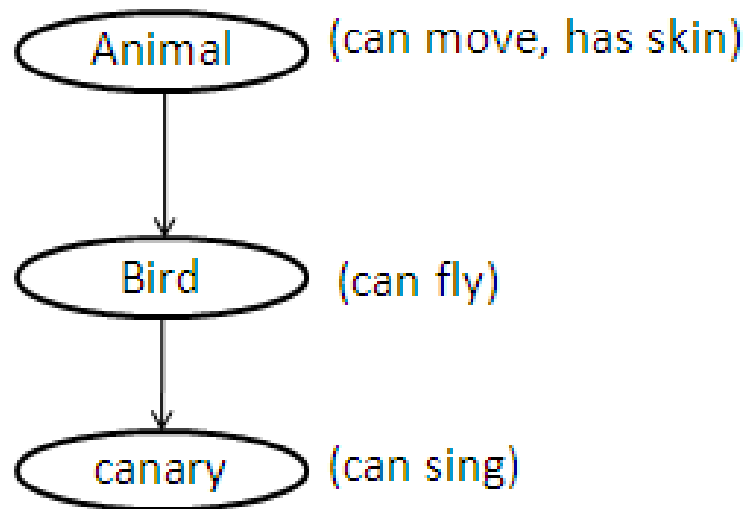
WordNet

What is a WordNet?

- WordNet is a lexical (related to vocabulary of a language) database for the language or network of words
- It organizes lexical information in terms of word meaning (context) rather than word form
- **Used for word sense disambiguation**
- Concept is search in a WordNet rather than word
- A big dictionary which can provide word meaning, part of speech, synonym, antonym and many more.

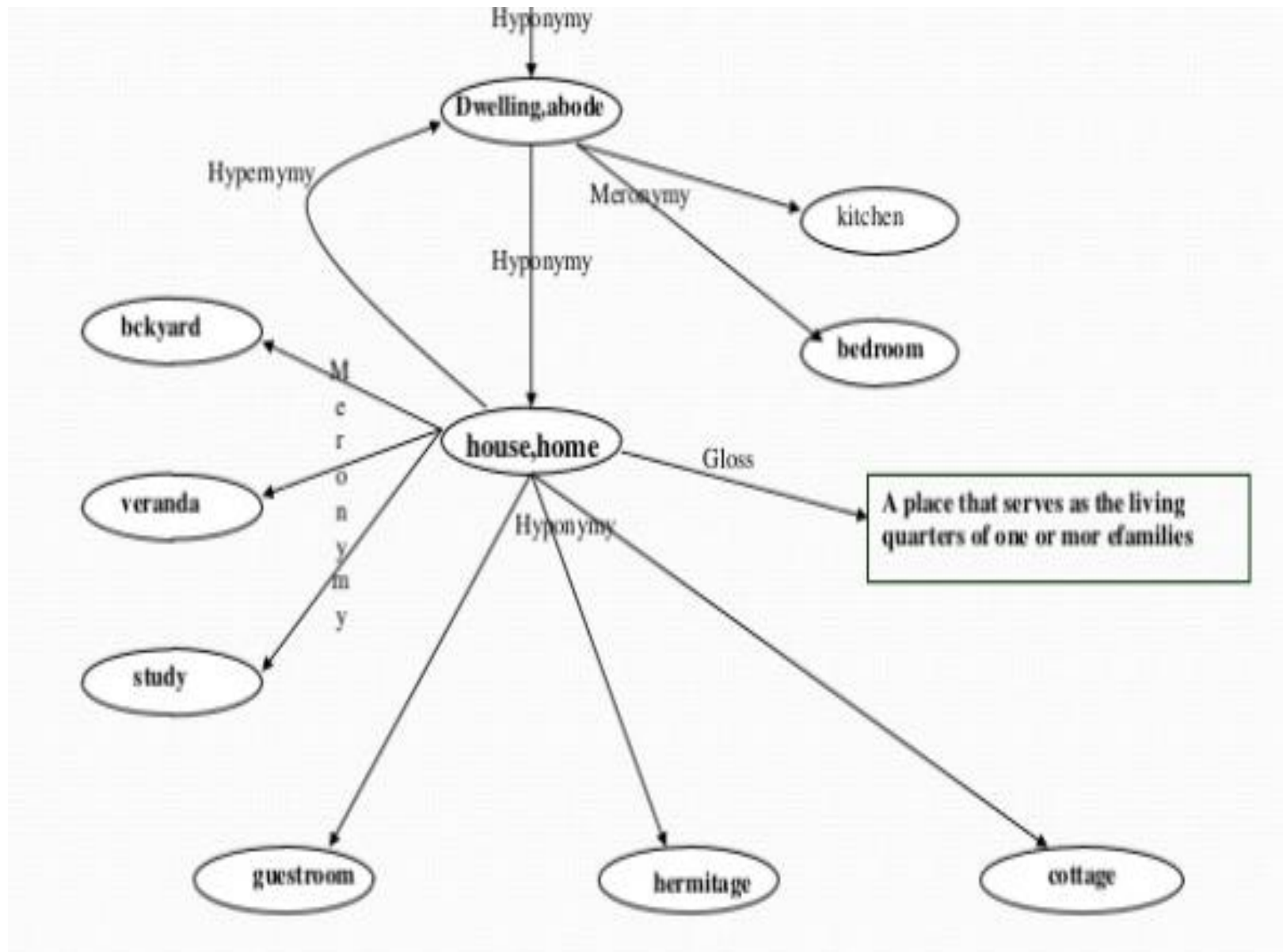
WordNet Concepts

- Human lexical memory for nouns as a hierarchy.
- *Can canary sing?* - *Pretty fast response.*
- *Can canary fly?* - *Slower response.*
- *Does canary have skin?* – *Slowest response.*



Wordnet - a lexical reference system based on psycholinguistic theories of human lexical memory.

WordNet Sub Graph



How is sense defined in WordNet

- The **synset (synonym set)**, the set of near-synonyms, instantiates a sense or concept, with a **gloss**
- Example: **chump** as a noun with the **gloss**:
“a person who is gullible and easy to take advantage of”
- This sense of “chump” is shared by 9 words:
chump¹, fool², gull¹, mark⁹, patsy¹, fall guy¹,
sucker¹, soft touch¹, mug²
- Each of **these** senses have this same gloss
 - (Not **every** sense; sense 2 of gull is the aquatic bird)

Words and synsets in Python

```
from nltk.corpus import wordnet  
Synarray=wordnet.synsets('dog')  
print(Synarray)
```

```
[ Synset('dog.n.01'), Synset('frump.n.01'), Synset('cad.n.01'),  
Synset('frank.n.02'), Synset('paw1.n.01'), Synset('andiron.n.01'),  
Synset('chase.v.01')]
```

- Each synset is labeled with a 3-part name of the form <word>.<pos>.<number>
- <word> is the word, <pos> is the part-of-speech of the word, <number> is the index of the sense. For example, 'dog.n.01' means the first meaning of 'dog' used as a noun.

Words and synsets in Python

- In order to focus only on the senses of a word used as a particular part-of-speech specify in the synsets().

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

```
wordnet.synsets('dog','n')
```

- returns the list of senses of 'dog' used as a noun:

```
[Synset('dog.n.01'), Synset('frump.n.01'), Synset('cad.n.01'),  
Synset('frank.n.02'), Synset('paw1.n.01'),  
Synset('andiron.n.01')]
```

- The following returns the list of senses of 'dog' used as a verb:

```
wordnet.synsets('dog','v')
```

```
[Synset('chase.v.01')]
```

- The two other parts-of-speech we can use are 'a' and 'r', short for adjectives and adverbs respectively.

Lemmas in Python

- A *lemma* is wordnet's version of an entry in a dictionary: A word in basic form, with a single meaning.
- Each synset contains one or more lemmas, which represent a specific sense of a specific word.
- The words that belong to a synset are called *lemmas in WordNet*. We can refer to the lemmas of a synset by invoking the lemmas attribute.

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

```
wordnet.synset('dog.n.01').lemmas()
```

Output: [Lemma('dog.n.01.dog'), Lemma('dog.n.01.domestic_dog'),
Lemma('dog.n.01.Canis_familiaris')]

- The first three fields in lemma() point to the synset,
- Last field is the canonical form of the lemma that belongs to the synset

Sense relations

- Homonyms: Same sound and/or spelling . For Example bank (financial institute), bank (river)
- Hypernyms; X is a hypernym of Y if Y is a member of X. For example, 'dog' is a hypernym of 'poodle'. More abstract
- Hyponyms: X is a hyponym of Y if X is a member of Y. For example, 'poodle' is a hyponym of 'dog'. More specific.
- Holonyms : X is a holonym of Y if Y is a part of X. For example, 'bicycle' is a holonym of 'chain'.
- Meronyms: X is a meronym of Y, if X is a part of Y: X is one of the components that make up a single entity Y. For example, 'chain' is a meronym of 'bicycle'

Sense relations

- Synonyms: Two words are synonyms if their meanings are similar in the sense that replacing one by the other in a sentence does not change the meaning of the sentence
- Antonyms: Two words are antonyms if their meanings are "opposite" in the following senses:
 - (a) The two words show binary opposition: present vs. absent, for example.
 - (b) The two words are near the opposite ends of a spectrum: tall vs. short, for example.
 - (c) The two words express change or movement in opposite directions: rise vs. fall, for example

Sense relations in Python

Sense relations are captured by methods specific to lemmas and synsets. Below is a list of useful methods. Each method returns a list. If no related senses are found, we get an empty list.

<lemma>.antonyms

Antonyms of lemma

e.g. `wordnet.lemma('present.a.02.present').antonyms()`

<synset>.hypernyms

Hypernyms of synset

`wordnet.synset('dog.n.01').hypernyms()`

<synset>.hyponyms

Hyponyms of synset

`wordnet.synset('dog.n.01').hyponyms()`

Sense relations in Python

<synset>.member_holonyms

Groups consisting of the specified members

```
wordnet.synset('copilot.n.1').member_holonyms()
```

<synset>.member_meronyms

Members of the specified group

```
wordnet.synset('faculty.n.2').member_meronyms()
```

<synset>.substance_holonyms

Things made of the specified substance

```
wordnet.synset('gin.n.1').substance_holonyms()
```

Sense relations in Python

<synset>.part_holonyms

Things consisting of the specified parts

```
wordnet.synset('course.n.7').part_holonyms()
```

<synset>.part_meronyms

Parts of the specified whole

```
wordnet.synset('table.n.2').part_meronyms()
```

<synset>.attributes

List of synsets that describes the attributes of synset

```
wordnet.synset('black.a.01').attributes()
```

Complete Example

```
from nltk.corpus import wordnet
syns = wordnet.synsets("students")
print(syns[0].name())
print(syns[0].lemmas()[0].name())
print(syns[0].definition())
synonyms = []
antonyms = []
for syn in wordnet.synsets("students"):
    for l in syn.lemmas():
        synonyms.append(l.name())
        if l.antonyms():
            antonyms.append(l.antonyms()[0].name())
print(set(synonyms))
print(set(antonyms)) syn[0].examples()
```

Output

student.n.0l

student

a learner who is enrolled in an educational institution

set([u'educatee', u'scholar', u'bookman', 'scholarly_person',
u'pupil', u'student'])

set([])

Similarity Scores

We can also easily use WordNet to compare the similarity of two words and their tenses

```
w1 = wordnet.synset('ship.n.01')
```

```
w2 = wordnet.synset('boat.n.01')
```

```
print(w1.wup_similarity(w2))
```

Output: 0.9090909090909091

```
w1 = wordnet.synset('ship.n.01')
```

```
w2 = wordnet.synset('car.n.01')
```

```
print(w1.wup_similarity(w2))
```

Output: 0.6956521739130435

References

- <https://www.youtube.com/watch?v=onVfWqRO-Gc>