Applied Natural Language Processing

Dr Jeff Mitchell, University of Sussex

Autumn 2025

Where are we?

Previously:

- document classification
- document similarity

This time:

- lexical semantics
  - word meaning
  - ambiguity and variation

Lexical Semantics

Lexical Semantics

- Most NLE applications are concerned with **semantics i.e., meaning**
- Imagine if you had to decide the topic of documents written in an alien language which you had never seen before?
- Hopefully, the aliens will provide you at least with a dictionary:
  - which specifies the meaning of words (lexemes)
  - in a language you understand?
  - or at least specifies which words mean similar things so that you can
  - ***Infer that two documents mean similar things because they contain words with similar meanings***
- In NLE, providing computers with a richer representation of lexical semantics (i.e., word meaning) should lead to more intelligent applications

Stop and think

- What is the meaning of the word “plant”?
- How many different meanings can you think of?

Word senses

- Words are often **ambiguous**
- Words can have **multiple senses** (i.e., meanings)

*I placed the book on the **counter**.*

vs

*I placed my **counter** on the gameboard.*

- How many more senses of ***counter*** can you think of?

Dictionaries

- Lexicographers produce dictionaries which:
  - enumerate the senses of all the words in a language
  - provide definitions of different sense
  - provide examples of usage of different senses

Machine readable dictionaries

- Many dictionaries now available online
- Some even have an API or are available in a machine readable form.

WordNet online search:

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

Oxford English Dictionary online search:

https://en.oxforddictionaries.com/?utm_source=od-panel&utm_campaign=en

Accessing WordNet via Python NLTK

```
[3]  from nltk.corpus import wordnet as wn
     wn.synsets("plant")
```

```
[Synset('plant.n.01'),
 Synset('plant.n.02'),
 Synset('plant.n.03'),
 Synset('plant.n.04'),
 Synset('plant.v.01'),
 Synset('implant.v.01'),
 Synset('establish.v.02'),
 Synset('plant.v.04'),
 Synset('plant.v.05'),
 Synset('plant.v.06')]
```

The wordnet synsets() function takes a word as its argument and returns a list of all of the synsets which that word is part of. Each synset corresponds to a different sense of the word.

How many different senses do words have?

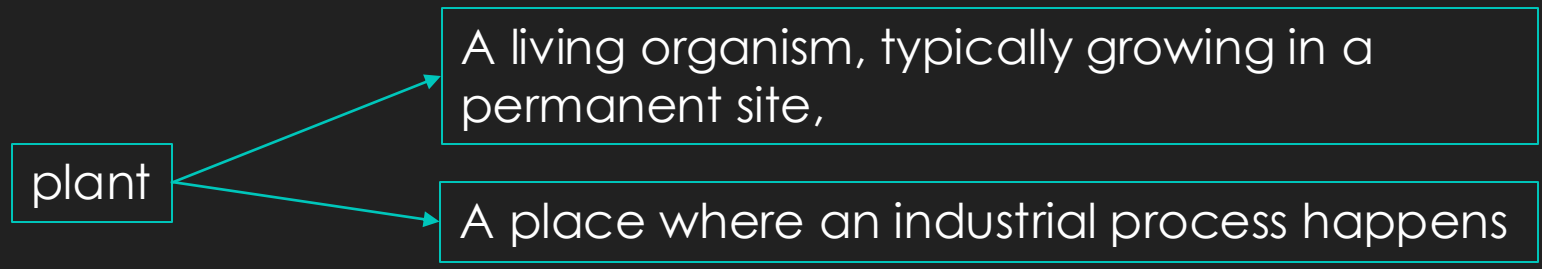
	WordNet	Oxford
plant	Noun:4, Verb:6	N:6, V:11
chicken	Noun:4, AdJ:1	N:4, V:1, J: 1
book	Noun: 11, Verb:4	N:14,V:9
twig	Noun:1, Verb: 2	N:2
counter	N: 9, V:2, J:1, R: 1	N:13, V: 3, J: 1, R: 1

Dictionaries do not always agree on this! Why is it so difficult?

J=adJective

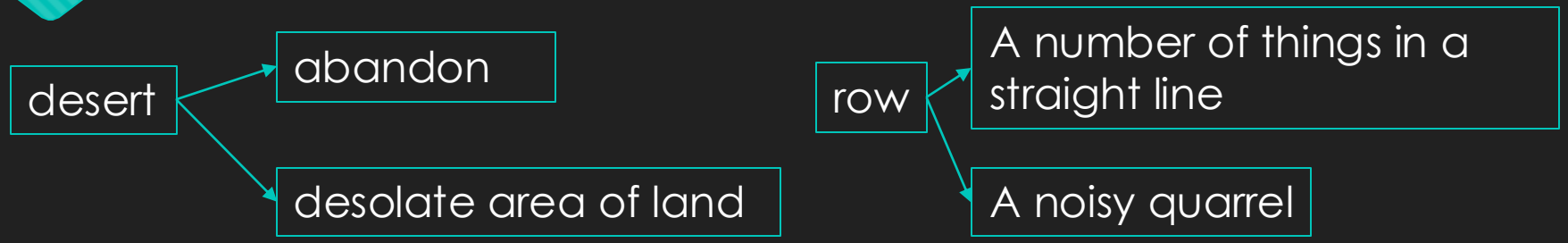
R=adverRb

Homonyms



- **Homonym** literally means "*same name*" (from Greek)
- Refers to **very broad differences in meaning**
- Different concepts appear to just happen to have been labelled with the same word form
- May be due to different etymology (e.g., one meaning comes from Anglo-Saxon and another from French or Latin)
- Or the derivational process is long forgotten

Homographs



- **Different senses can have different pronunciations**
- Literally means "*same writing*"
- Less of a problem when dealing with speech input data
- Big problem in text-to-speech applications

Homophones

there vs their

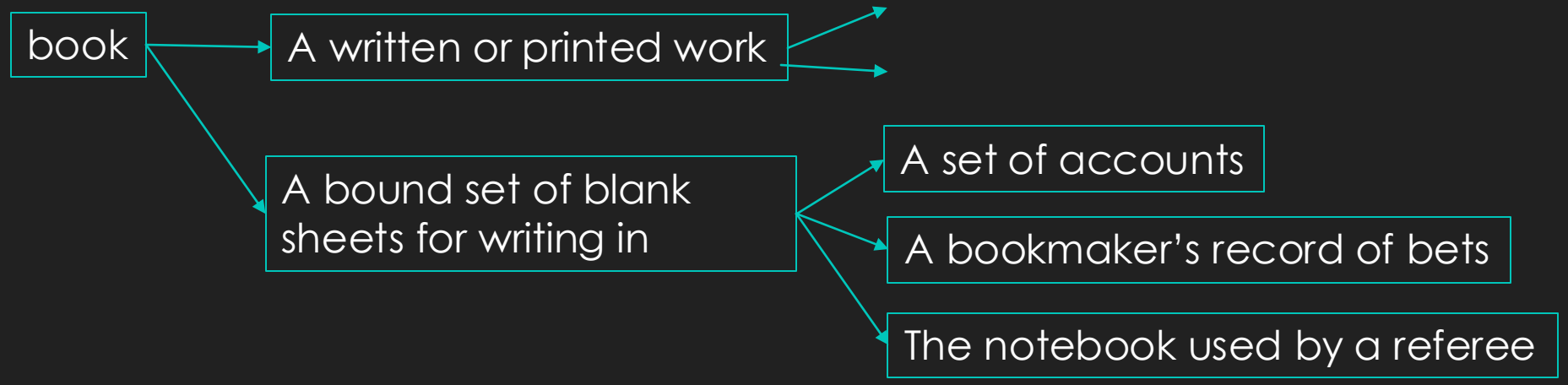
bear vs bare

new vs knew

blue vs blew

- Different meanings can have word forms with the **same pronunciation** but **different spellings**
- Literally means "*same sound*"
- Big problem for speech-input applications
- Less of a problem for text or text-to-speech applications

Polysemy



- Literally means "*many senses*"
- Possible to make many fine-grained sense distinctions
- Senses are often related

Getting Definitions of Senses from WordNet

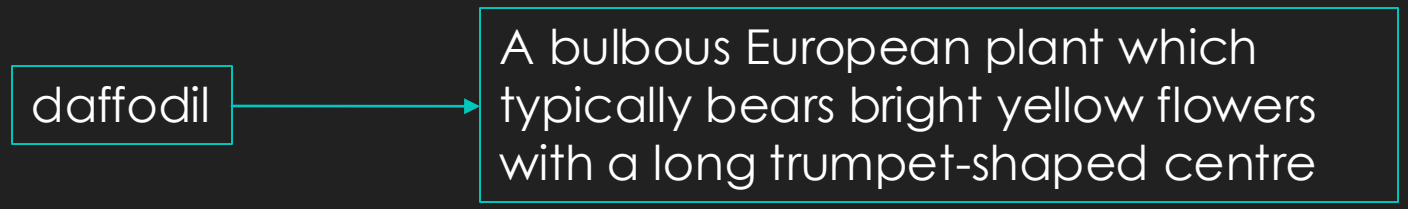
```
plant_synsets=wn.synsets('plant')
for i,s in enumerate(plant_synsets):
  print("{} {}".format(i+1,s.definition()))
```

The **synset** class has a definition method

```
1 buildings for carrying on industrial labor
2 (botany) a living organism lacking the power of locomotion
3 an actor situated in the audience whose acting is rehearsed but seems spontaneous to the audience
4 something planted secretly for discovery by another
5 put or set (seeds, seedlings, or plants) into the ground
6 fix or set securely or deeply
7 set up or lay the groundwork for
8 place into a river
9 place something or someone in a certain position in order to secretly observe or deceive
10 put firmly in the mind
```

Which of these senses of 'plant' do you think are related?

Monosemous words



- Literally means "*single sense*"
- Very few truly monosemous words in English
- Although some words do have a strongly predominant sense;
- or might be considered monosemous in a restricted domain

Semantic Relationships

WordNet

- More than an electronic dictionary!
- See http://wordnet.princeton.edu for more general information
- Or see: Christiane Fellbaum (1998, ed.) *WordNet: An Electronic Lexical Database*. MIT Press.

Relationships between word senses

quickly/R

poodle/N

dog/N

move/V

red/J

walk/V

sleek/J

house/N

saddle/N

slowly/R

hot/J

horse/N

love/V

animal/N

cold/J

hoof/N

fast/R

Semantic relationships

- synonymy
- antonymy
- hyponymy / hypernymy

Synonymy

fast == quickly

- Words which mean the same thing
- *Two words are **synonymous** if they can be substituted in all possible contexts without changing the meaning of the utterance.*
- True synonyms are very rare
- Choice of synonym usually gives us some extra information about the situation or speaker e.g., *car* vs *automobile*
- It is often defined as a relationship between word senses rather than between words. e.g., *plant == spy* ?

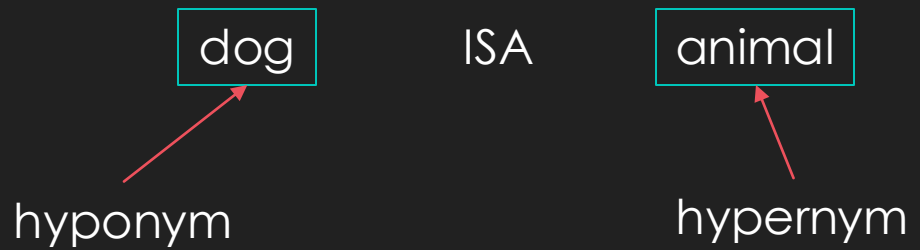
Antonymy

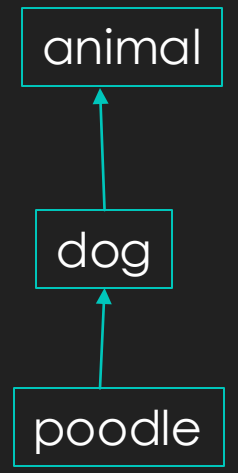
hot ≠ cold

- Words which are opposite in meaning
- Substituting one for the other would often cause a contradiction:
  - *The food is hot*.
  - *The food is cold*.
- Antonyms are actually very similar in meaning
  - *hot* and *cold* both describe the temperature of an object
  - *rise* and *fall* both describe an object which is moving in the vertical plane
- Most antonym pairs are adjectives, verbs or adverbs

Hyponymy and Hypernymy

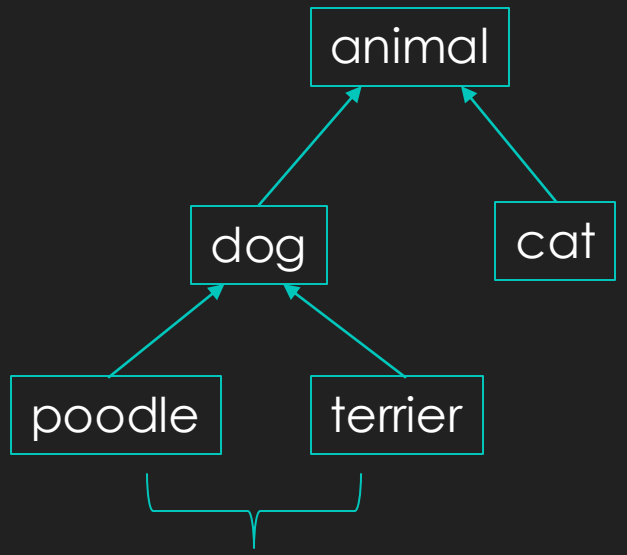
- Linguistic terms which capture the idea of class inclusion
- A *dog* is a type of *animal* so:
  - *dog* is a **hyponym** of *animal*
  - *animal* is a **hypernym** of *dog*
- It's a transitive relationship so
  - If *dog* is a hyponym of *animal*
  - And *poodle* is a hyponym of *dog*
  - *Poodle* is also a hyponym of *animal*





Hyponym Hierarchies

- The hyponymy relationship links together large numbers of concepts in a tree or hierarchy
- Most general superclass at the top
- Most specific types at the leaves



Words which share a common hypernym are called **co-hyponyms**

WordNet

- A linguistic network organized around synonymy and hyponymy
- Core unit is the **synset**
  - a set of synonymous word senses
  - a set may contain a single word
  - synset items may be bigrams (e.g., "plant life") as well as unigrams
  - each synset is also associated with a single definition
- Polysemous words appear in multiple synsets
  - One for each sense
- Synsets are then connected via hyponymy…..

{**plant, flora, plant life**} = a living organism lacking the power of locomotion

{plant} = something planted secretly for discovery by another

{**plant, works, industrial plant**} = buildings for carrying on industrial labour

Synonyms in WordNet NLTK

```
cat_synsets = wn.synsets("cat",wn.NOUN)
for i,s in enumerate(cat_synsets):
    wordforms=[l.name() for l in s.lemmas()]
    print("{}:{}\n\t{}".format(i,wordforms,s.definition()))
```

A Lemma is a class which can be thought of as a sense. We use its name() method to get the word form itself.

```
0:['cat', 'true_cat']
        feline mammal usually having thick soft fur and no ability to roar: domestic cats; wildcats
1:['guy', 'cat', 'hombre', 'bozo']
        an informal term for a youth or man
2:['cat']
        a spiteful woman gossip
3:['kat', 'khat', 'qat', 'quat', 'cat', 'Arabian_tea', 'African_tea']
        the leaves of the shrub Catha edulis which are chewed like tobacco or used to make tea; has the effect of a e
uphoric stimulant
4:["cat-o'-nine-tails", 'cat']
        a whip with nine knotted cords
5:['Caterpillar', 'cat']
        a large tracked vehicle that is propelled by two endless metal belts; frequently used for moving earth in con
struction and farm work
6:['big_cat', 'cat']
        any of several large cats typically able to roar and living in the wild
7:['computerized_tomography', 'computed_tomography', 'CT', 'computerized_axial_tomography', 'computed_axial_tomograph
y', 'CAT']
        a method of examining body organs by scanning them with X rays and using a computer to construct a series of
cross-sectional scans along a single axis
```

Hypernyms in WordNet NLTK

Remember **cat_synsets** is a list of Synset objects. The Synset class has a hypernyms() method which returns another list of Synset objects ….

```
for h in cat_synsets[6].hypernyms():
    h_words=[w.name() for w in h.lemmas()]
    print("{}:{}".format(h_words,h.definition()))
```

```
['feline', 'felid']:any of various lithe-bodied roundheaded
fissiped mammals, many with retractile claws
```

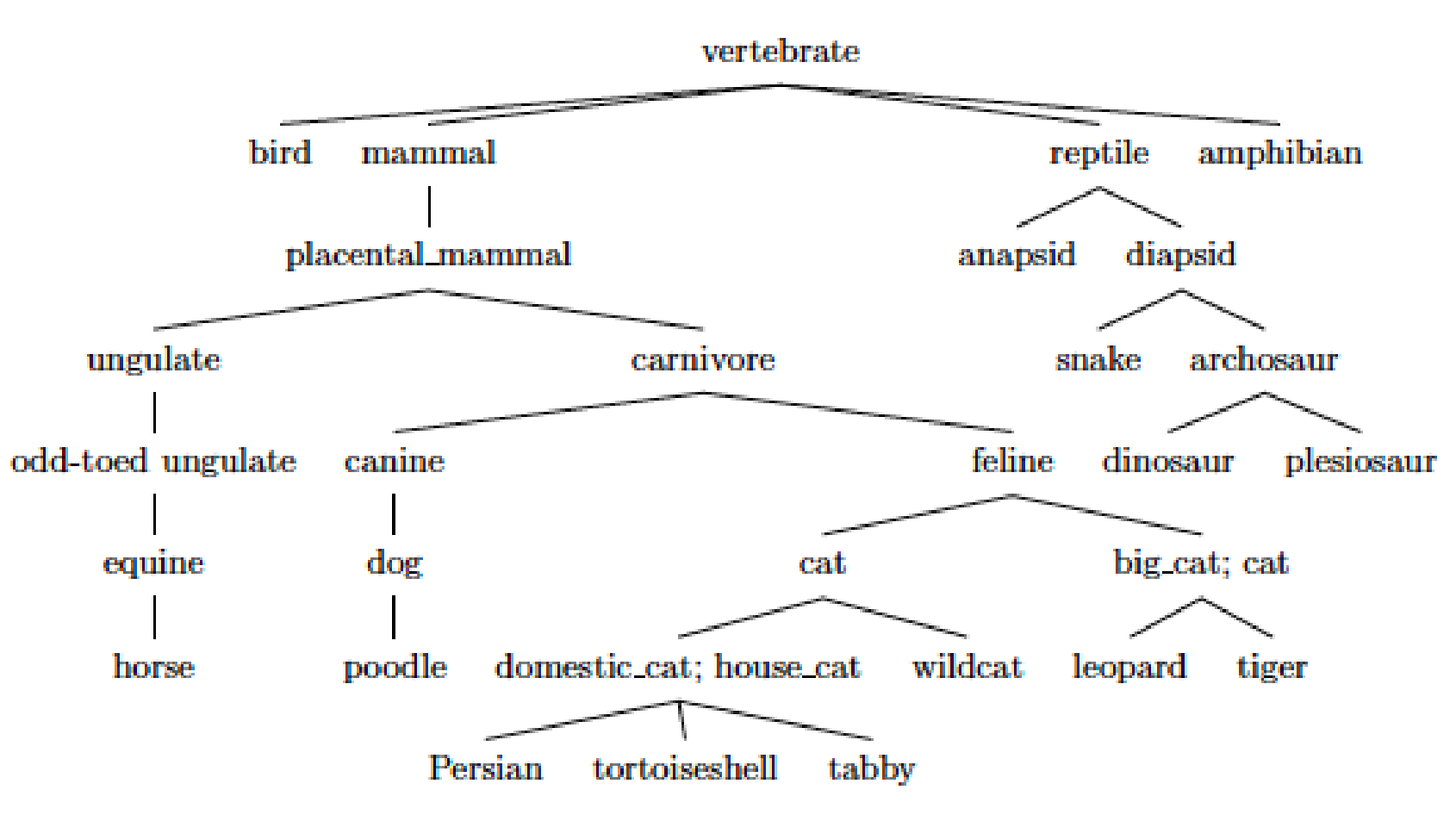
The 6th sense of cat has a single hypernym, commonly referred to as 'feline' or 'felid

Hyponyms in WordNet NLTK

```
for h in cat_synsets[6].hyponyms():
    h_words=[w.name() for w in h.lemmas()]
    print("{}:{}".format(h_words,h.definition()))
```

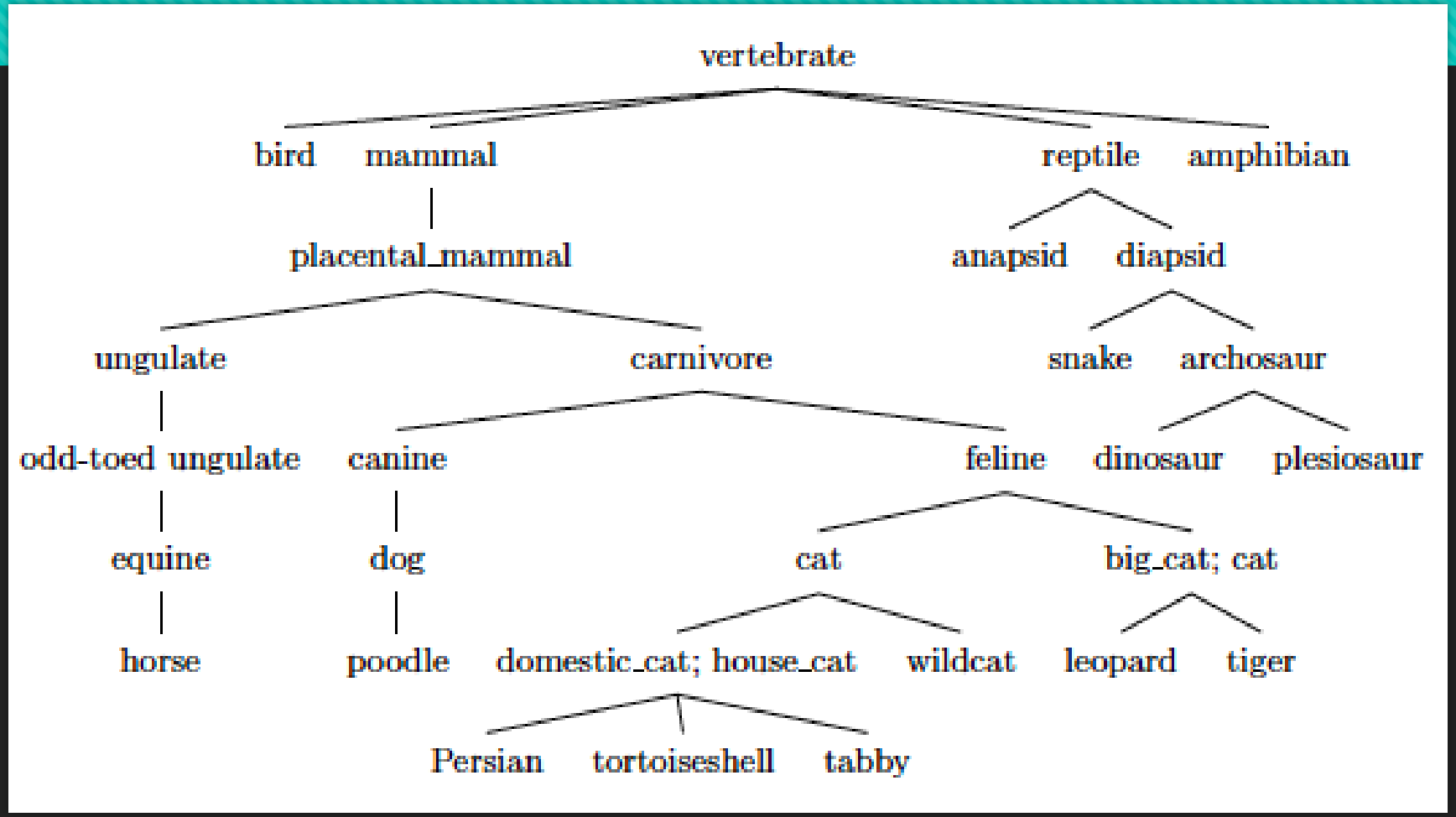
```
['cheetah', 'chetah', 'Acinonyx_jubatus']:long-legged spotted cat of Afri
ca and southwestern Asia having nonretractile claws; the swiftest mammal;
can be trained to run down game
['jaguar', 'panther', 'Panthera_onca', 'Felis_onca']:a large spotted feli
ne of tropical America similar to the leopard; in some classifications co
nsidered a member of the genus Felis
['leopard', 'Panthera_pardus']:large feline of African and Asian forests
usually having a tawny coat with black spots
['liger']:offspring of a male lion and a female tiger
['lion', 'king_of_beasts', 'Panthera_leo']:large gregarious predatory fel
ine of Africa and India having a tawny coat with a shaggy mane in the mal
e
['saber-toothed_tiger', 'sabertooth']:any of many extinct cats of the Old
and New Worlds having long swordlike upper canine teeth; from the Oligoce
ne through the Pleistocene
['snow_leopard', 'ounce', 'Panthera_uncia']:large feline of upland centra
l Asia having long thick whitish fur
['tiger', 'Panthera_tigris']:large feline of forests in most of Asia havi
ng a tawny coat with black stripes; endangered
['tiglon', 'tigon']:offspring of a male tiger and a female lion
```

Extract from the WordNet noun hierarchy



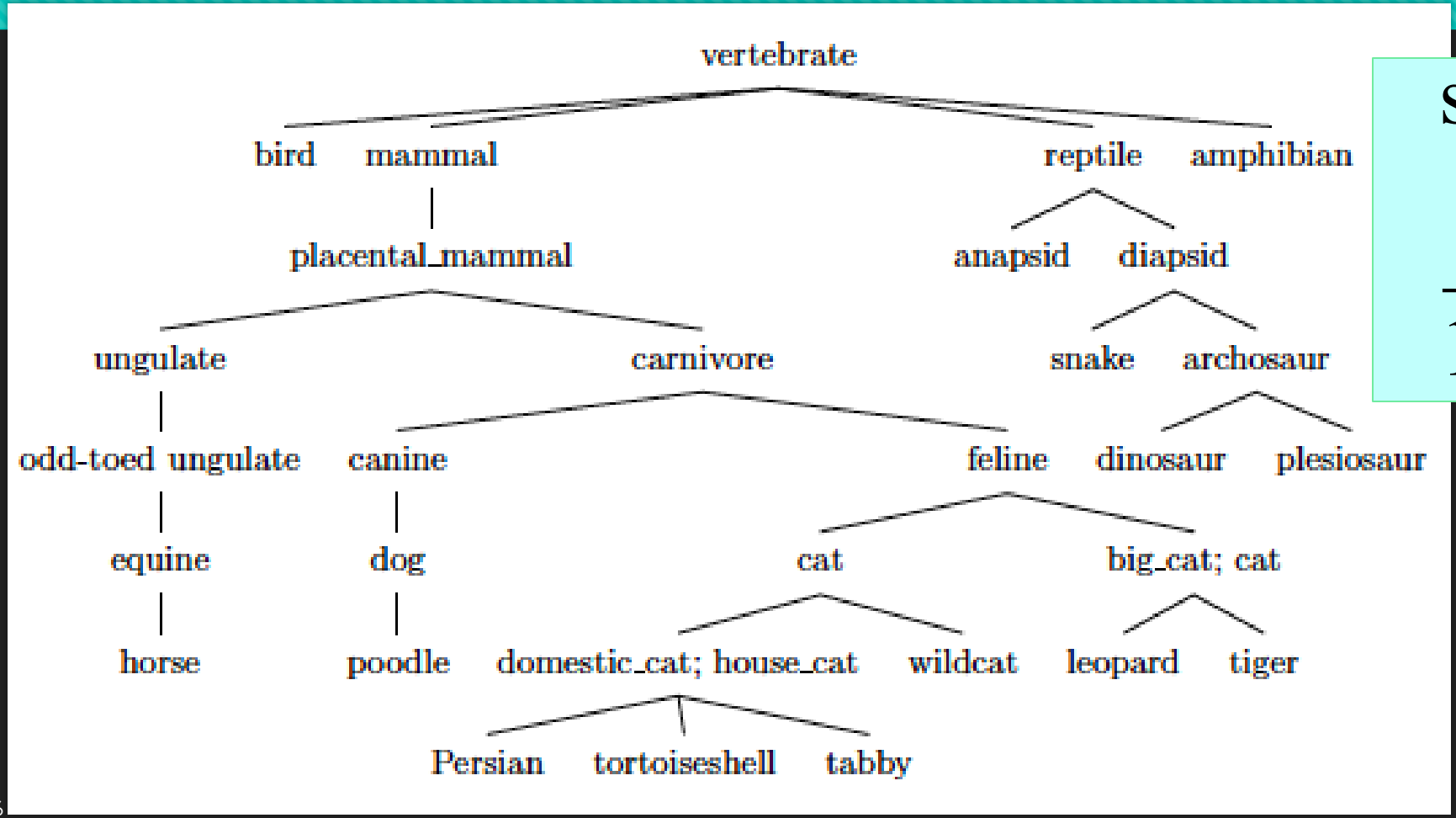
Semantic Similarity

Semantic similarity based on WordNet



Intuition: More similar concepts are closer together in the hierarchy.

Path length: shorter path -> greater similarity



$$\mathrm{sim}_{\mathrm{path}}(c_1, c_2) = \frac{1}{1 + \mathrm{pathlen}(c_1, c_2)}$$

Stop and think

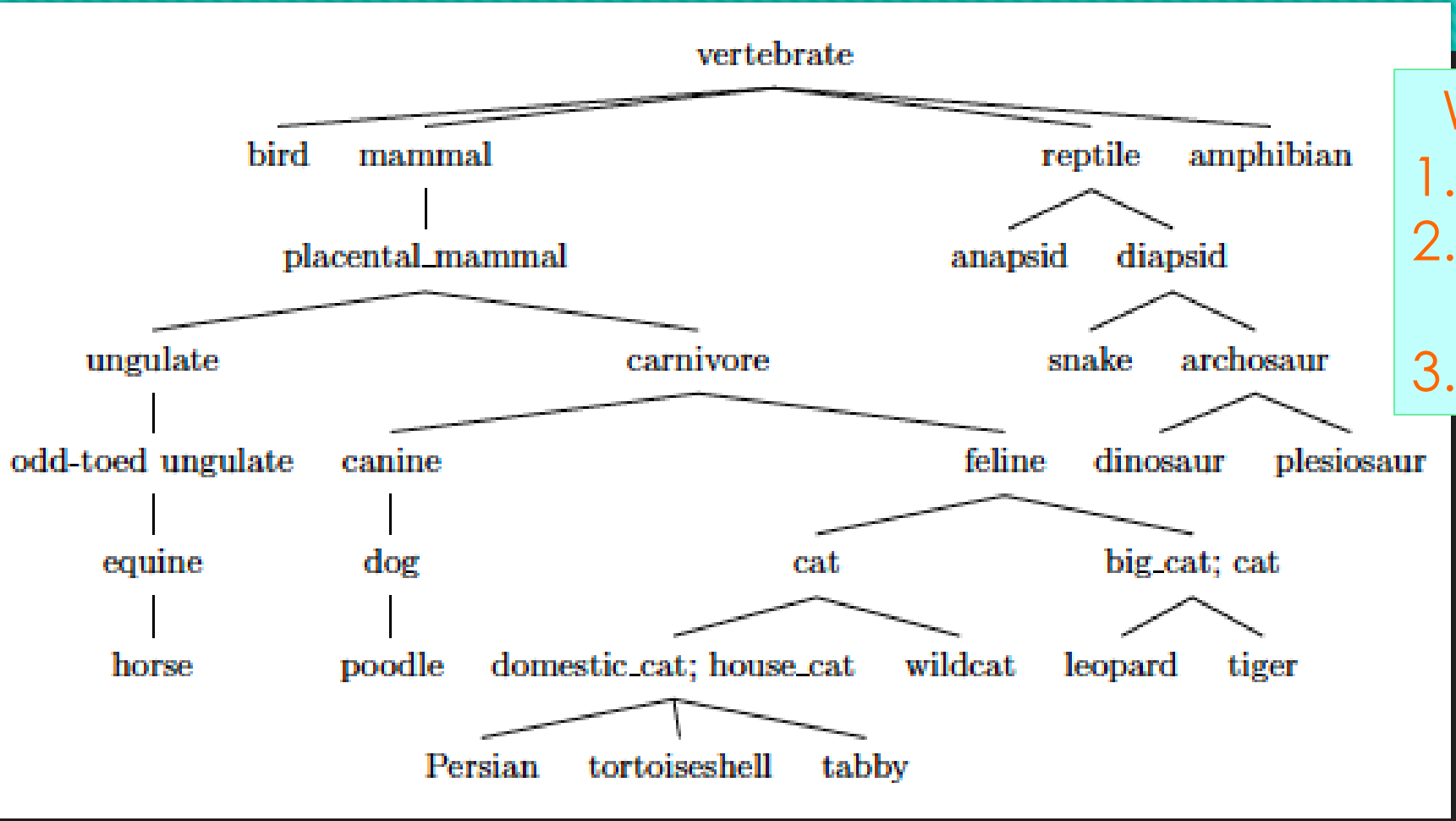
- How many pairs of words can you find which have a path length similarity score of 1/5 according to the extract of the WordNet hierarchy shown on the previous slide?

Potential problems with pathlength

- Pathlength does not differentiate between different types of path e.g., *canine* → … → *vertebrate* vs *dog* → … ← *cat*
- Intuitively, concepts (separated by same path length) are more dissimilar higher up the tree; but this is not captured by path length similarity measure
- Some parts of tree may be densely populated with rare terminology

Lowest common subsumer (LCS):

similarity based on what two concepts share



What is the LCS of:

1. tabby and tiger?
2. poodle and carnivore?
3. poodle and tiger?

Information content

- Intuition: concepts which have the LCS *carnivore* are more similar than concepts which have the LCS *vertebrate*
- We gain more **information** when we are told two objects are both *carnivores* than when we are told they are both *vertebrates*.
- We capture this probabilistically via ***the information content (IC) of a concept***
  - Annotate the hierarchy with the frequency of occurrence of each concept in some corpus
  - Remember that the occurrence of a concept implies the occurrence of all of its hypernyms (if something is a *dog*, it is also a *canine* and so on)

$$P(c) = \frac{\text{freq}(c)}{\sum_c \text{freq}(c)}$$

$$IC(c) = -\log(P(c))$$

WordNet similarity measures based on information content (IC)

$$\mathrm{IC}(c) = -\log P(c)$$

Information content in a concept

$$\mathrm{sim}_{\mathrm{res}}(c_1, c_2) = \mathrm{IC}(\mathrm{LCS}(c_1, c_2))$$

Information content in what the concepts share (their lowest common subsumer)

See Resnik, 1995

$$\mathrm{sim}_{\mathrm{lin}}(c_1, c_2) = \frac{2 \times \mathrm{sim}_{\mathrm{res}}(c_1, c_2))}{\mathrm{IC}(c_1) + \mathrm{IC}(c_2)}$$

Ratio of shared information content to total information content

See Lin 1998b

Word similarity

$$\text{wordsim}(w_1, w_2) = \max_{\substack{c1 \in senses(w_1) \\ c2 \in senses(w_2)}} \text{sim}(c_1, c_2)$$

- You will be writing python code to implement this function in the lab exercises.

Evaluation

- How do we evaluate these measures? What is the right answer?

Human synonymy judgements

- Rubenstein & Goodenough 1965 (65 pairs)
- Miller and Charles 1991 (30 pairs)
- WordSim-353 2002 (353 pairs)
- MEN dataset 2012 (3000 pairs)

	M&C	WN
car-automobile	3.92	1.00
magician-wizard	3.5	1.00
journey-car	1.16	0.00
coast-forest	0.42	0.15
noon-string	0.08	0.00

Humans rank the pair of words ***(car,automobile)*** as being the most similar pair in this set of 5 word pairs.

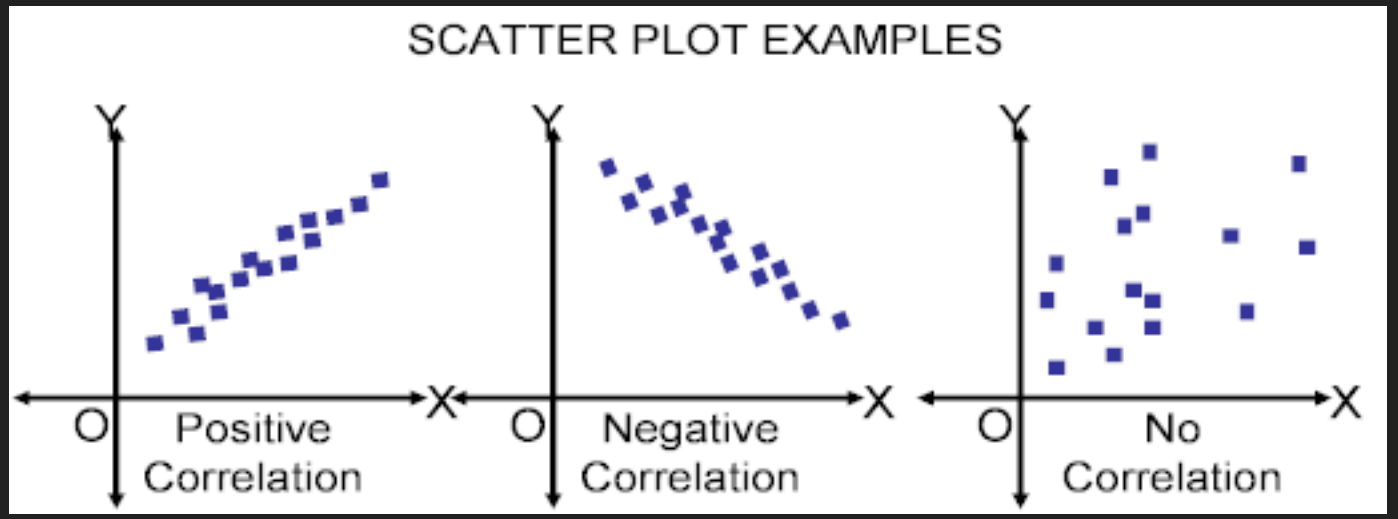
```
measures=["path","res","lin"]
for measure in measures:
    scores=[]

    for triple in mcdata:
        scores.append(word_similarity(triple[0],triple[1],measure=measure))
    df[measure]=scores

df
```

	word1	word2	human similarity	path	res	lin
0	asylum	madhouse	3.61	0.500000	9.475167	0.855584
1	bird	cock	3.05	0.500000	7.677755	0.773937
2	bird	crane	2.97	0.250000	7.677755	0.747812
3	boy	lad	3.76	0.500000	8.399492	0.830562
4	brother	monk	2.82	0.500000	9.261593	0.986407

Correlation

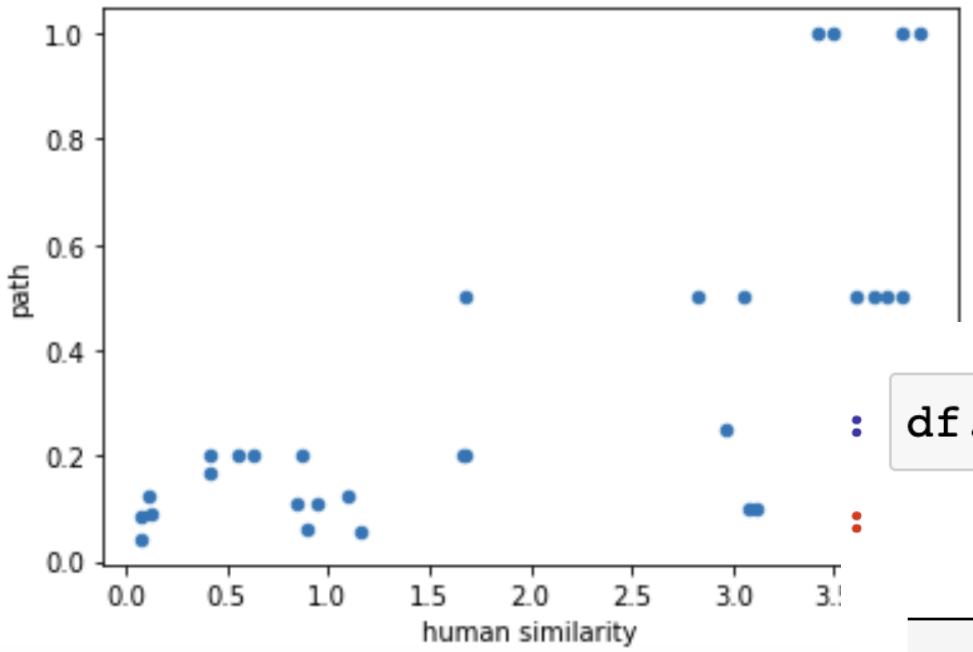


- Pearson's product-moment correlation coefficient
- Spearman's rank correlation coefficient

```
x="human similarity"
y="path"

df.plot.scatter(x,y)
```

```
: <AxesSubplot:xlabel='human similarity', ylabel='path'>
```




```
: df.corr(method='spearman')
:
```

	human similarity	path	res	lin
human similarity	1.000000	0.722743	0.735945	0.753510
path	0.722743	1.000000	0.900648	0.945509
res	0.735945	0.900648	1.000000	0.962707
lin	0.753510	0.945509	0.962707	1.000000

Word Sense Disambiguation (WSD)

Example

pine cone

PINE:
1. Evergreen coniferous tree which has clusters of long needle-shaped leaves
2. Straight-grained durable and often resinous white to yellowish timber of a pine tree
3. Have a desire for something or someone not present
4. Waste away through sorrow or illness

CONE:
1. A solid or hollow object which tapers from a roughly circular base to a point which narrows to a point
2. The dry fruit of a conifer, typically tapering to a rounded end and formed of a tight array of overlapping scales on a central axis which separate to release the seeds
3. One of two types of light-sensitive cell in the retina of the eye, responding mainly to bright light and responsible for sharpness of vision and colour perception

What is the intended sense of each of the words in the phrase *pine cone*?

WSD – how do we know the sense of a word?

1. The willows lined the **bank** of the stream.
2. He skiied straight into a **bank** of snow.
3. The adult fish often seek out gravel **banks** in the shallows.
4. The Olympic competitors use the steep **banks** to reach top speed.
5. The early ships only had twenty-five oars in each **bank**.
6. I must go to the **bank** and change some money.

One Sense Per Collocation

- A ***collocation*** is a pair (or group) of words which frequently co-occur together in some defined relationship
- Can be defined as words co-occurring more often than one would expect by chance
- Relationship maybe based on adjacency / proximity or a syntactic relationship
- Yarowsky (1993) showed that, depending on definition of sense and collocation, an ambiguous word has only one sense in a given collocation with probability 90-99%

Collocation	Freq. as aid	Freq as aide
foreign	718	1
federal	297	0
western	148	0
provide	88	0
covert	26	0
oppose	13	0
future	9	0
similar	6	0
presidential	0	63
chief	0	40
longtime	0	26
aids-infected	0	2
sleepy	0	1
disaffected	0	1
indispensable	2	1
practical	2	0

Example: Typical collocational distribution for the homophone ambiguity aid/aide

One Sense Per Discourse

- Sense usages tend to "clump"
- Gale et al. (1992) demonstrated that there is a very strong tendency (98%) for multiple uses of the same word to share the same sense in a well-written coherent discourse.

This site, at Billingham in the north-east of England, is a good example of such an integrated chemical **plan**t. All the **plants** are interconnected by steam pipes to make the most efficient use of energy released during manufacturing processes.

Approaches to WSD

- Knowledge-based methods
- Supervised corpus-based methods
- Semi-supervised corpus-based methods
  - Bootstrapping
  - Active learning
- Unsupervised corpus-based methods

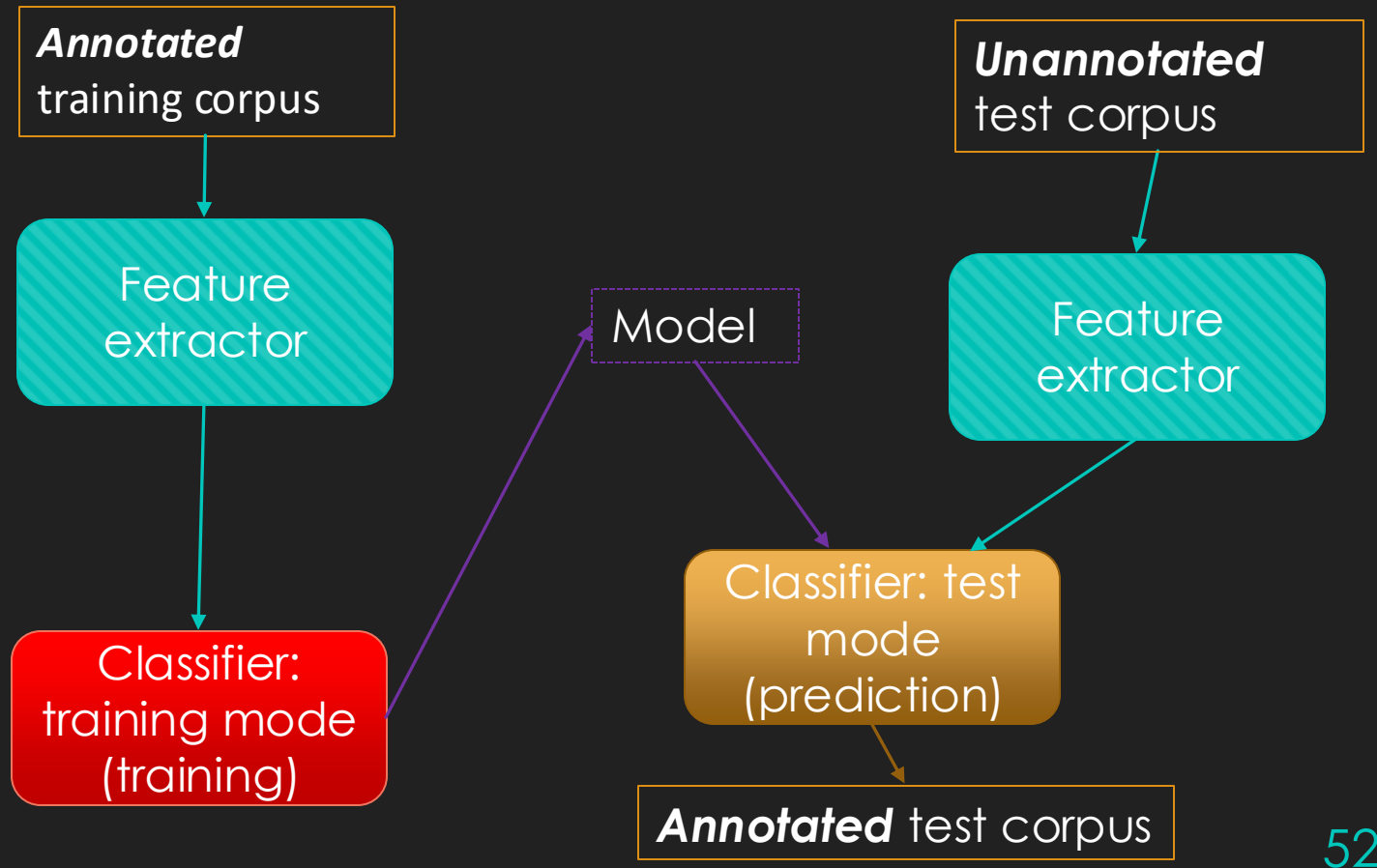
Knowledge-based methods

- Methods that rely primarily on dictionaries, thesauri and lexical knowledge bases without using any corpus evidence
- Also referred to as dictionary-based methods
- Include
  - Hand-crafted disambiguation rules
  - Comparing dictionary definitions to the context (Lesk algorithm)
  - Use of semantic similarity measures

Supervised Corpus-Based Methods

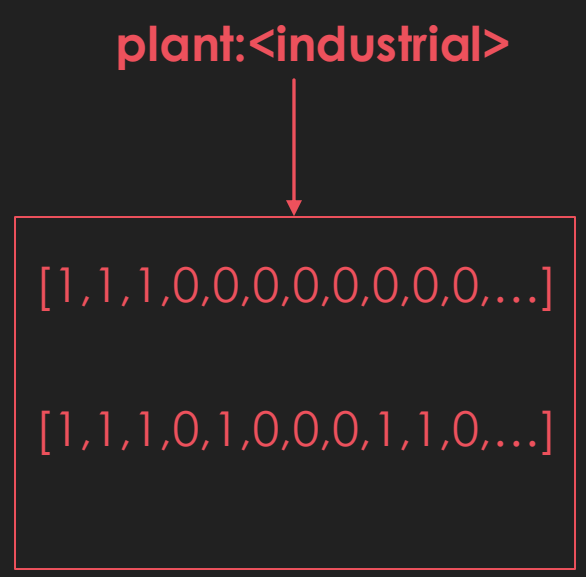
Require:

- Sense inventory i.e., a pre-specified set of *class* labels for each word of interest
- Training data i.e., a corpus of examples annotated with the class labels



Feature Extraction

- Codify each of the examples of a particular sense of a word as a *feature vector*
- Typically, features are binary and indicate presence or absence of something from an example
- For example:
  - "Is tagged as a noun"
  - "Is tagged as a verb"
  - "word X is adjacent"
  - "word X is in a context window of 10"
  - "word X is in the document"



Classifiers

- Given the (codified) training examples, find a model which best predicts those training examples
- Possibilities include:
  - **Naïve Bayes (NB) classifier**
  - Logistic Regression / Maximum Entropy classifier
  - **Nearest neighbours (kNN)**
  - Support vector machine (SVM)
  - Neural network

Naïve Bayes Classifier (Leacock et al. 1998)

[1,1,1,0,0,0,0,1,0,1] → **plant**<?>

For a given test example, you want to predict the sense, s, which has maximum probability given its feature vector $\underline{f}$

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}} P(s|\underline{f})$$

Applying Bayes Rule

conditional probability

prior probability

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}} \frac{P\left(\underline{f}|s\right) \times P(s)}{P\left(\underline{f}\right)}$$

Ignore the denominator (this is independent of the sense)

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}} P\left(\underline{f}|s\right) \times P(s)$$

To estimate the conditional distribution $P\left(\underline{f}|s\right)$, make the **naïve** assumption that features are independent

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}} \prod_{j=1}^{n} P(f_j|s) \times P(s)$$

Maximum Likelihood Estimation (MLE)

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}} \prod_{j=1}^{n} P(f_j|s) \times P(s)$$

Probabilities of individual features given a sense can easily be estimated from the training data using MLE

Prior probabilities P(s), are easy to estimate from the training data using MLE

P(f | s) = proportion of occurrences of s in training data which have feature f

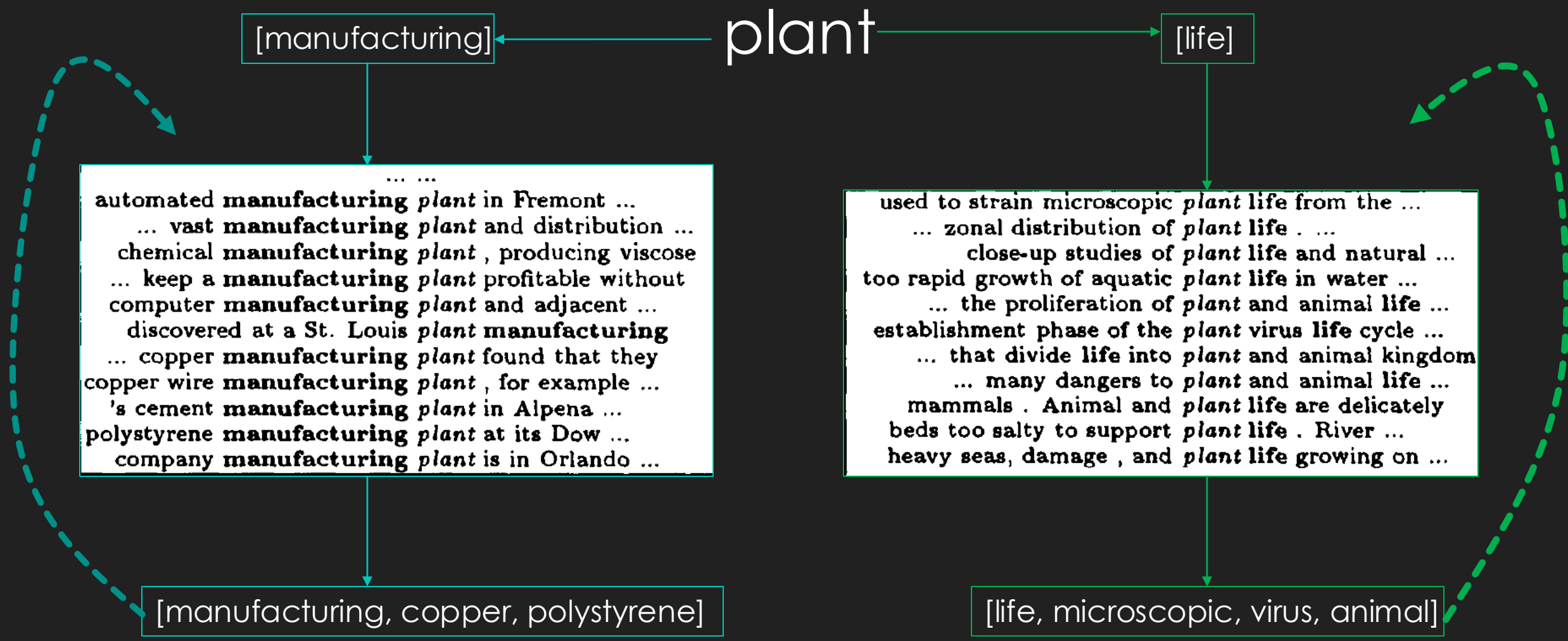
P(s) = proportion of occurrences of the target word in the training data which are labelled s

Semi-supervised corpus-based methods

- May be referred to as unsupervised because they no longer rely on an annotated training corpus
- However, generally referred to as semi-supervised because they still require:
  - Sense inventory i.e., a pre-specified set of *class* labels for each word of interest
  - Some expert input

Bootstrapping (Yarowsky, 1995)

1. Start with some seed collocates for each ambiguous word under consideration, which reliably disambiguate the word
2. Find examples in corpus where these collocates occur and tag them accordingly.
3. Use these tagged examples to find other collocates which reliably partition the tagged examples
4. REPEAT


plant
[manufacturing]
[life]
... ...
automated manufacturing plant in Fremont ...
... vast manufacturing plant and distribution ...
chemical manufacturing plant , producing viscose
... keep a manufacturing plant profitable without
computer manufacturing plant and adjacent ...
discovered at a St. Louis plant manufacturing
... copper manufacturing plant found that they
copper wire manufacturing plant , for example ...
's cement manufacturing plant in Alpena ...
polystyrene manufacturing plant at its Dow ...
company manufacturing plant is in Orlando ...
used to strain microscopic plant life from the ...
... zonal distribution of plant life . ...
close-up studies of plant life and natural ...
too rapid growth of aquatic plant life in water ...
... the proliferation of plant and animal life ...
establishment phase of the plant virus life cycle ...
... that divide life into plant and animal kingdom
... many dangers to plant and animal life ...
mammals . Animal and plant life are delicately
beds too salty to support plant life . River ...
heavy seas, damage , and plant life growing on ...
[manufacturing, copper, polystyrene]
[life, microscopic, virus, animal]

Stop and think

- Where might the initial seed collocations come from for the bootstrapping method?

Reliability of seed collocates

- Reliability usually measured in terms of:
  - Frequency of seed with sense A
  - Frequency of seed with sense B
- Seeds which have occurred frequently with one sense but not the other are more reliable
- Introducing less reliable seed collocates for a sense A may
  - increase recall (i.e., increase the number of instances of sense A which can be tagged correctly)
  - decrease precision (i.e., increase the number of instances of the other sense which are erroneously tagged as sense A
- May be possible to overcome some errors by applying one sense per discourse heuristic