COMP90042

Workshop Week 03

Workshops

- 10 sessions in total

Mon 11-12pm	Alice Hoy-108
Mon 6:15-7:15pm	Alice Hoy-108
Mon 7:15-8:15pm	Alice Hoy-108 *
Tues 10-11am	Alice Hoy-222 *
Tues 6:15-7:15pm	Alice Hoy-108 *
Wed 8-9am	Alice Hoy-109
Fri 1-2pm	Alice Hoy-222
Fri 3:15-4:15pm	Alice Hoy-210
Fri 5:15-6:15pm	Alice Hoy-211
Fri 6:15-7:15pm	Alice Hoy-222

Questions…

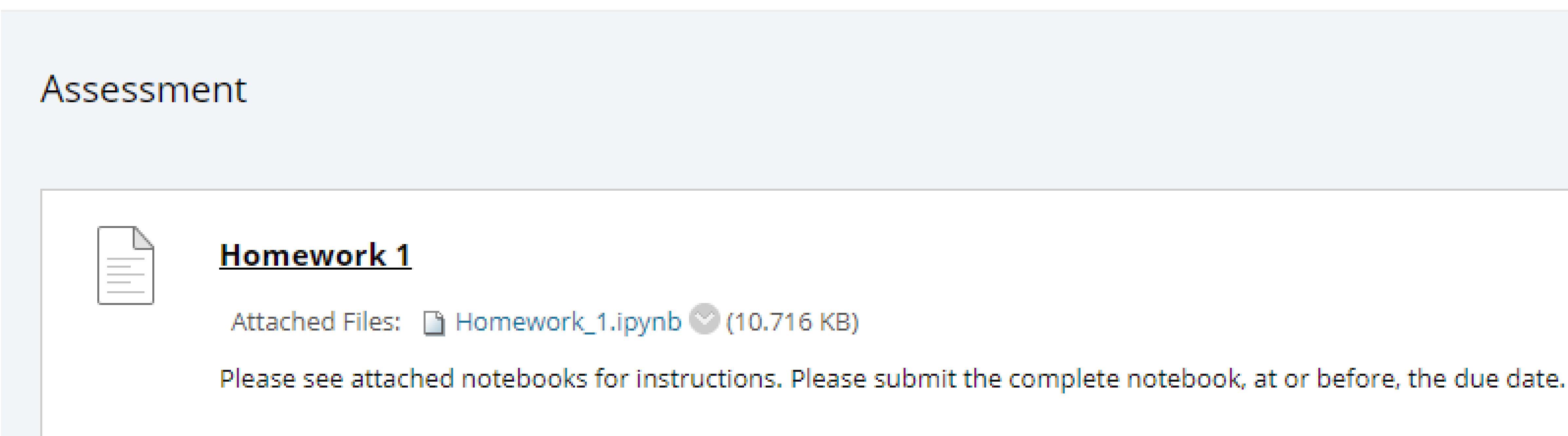
- Post on the LMS discussion board

- Trevor / Daniel
  - t.cohn@unimelb.edu.au / d.beck@unimelb.edu.au
  - Weekly office hour, Wed *12pm-1pm*, DMD 7.02 (new time)

- My contact
  - Yuan Li
  - yuanl4@student.unimelb.edu.au

Homework 1 released

- Due data: 11pm, Sunday March 18th
- We accept submissions written in Python 2.7 or 3.5
  - But 2.7 is still the recommended version
- LMS -> Assessment



When using code from notebooks…

```
lemmatizer = nltk.stem.wordnet.WordNetLemmatizer()

def lemmatize(word):
    lemma = lemmatizer.lemmatize(word,'v')
    if lemma == word:
        lemma = lemmatizer.lemmatize(word,'n')
    return lemma
```

Code from WSTA_N1B_preprocessing.ipynb

- According to Trevor's reply on LMS: … *indicate with comments what code is not original*, at the top and bottom of the snippet, and attribute the source clearly…

```
## Code below taken from WSTA_N1B_preprocessing.ipynb
lemmatizer = nltk.stem.wordnet.WordNetLemmatizer()

def lemmatize(word):
    lemma = lemmatizer.lemmatize(word,'v')
    if lemma == word:
        lemma = lemmatizer.lemmatize(word,'n')
    return lemma
## End of copied code
```

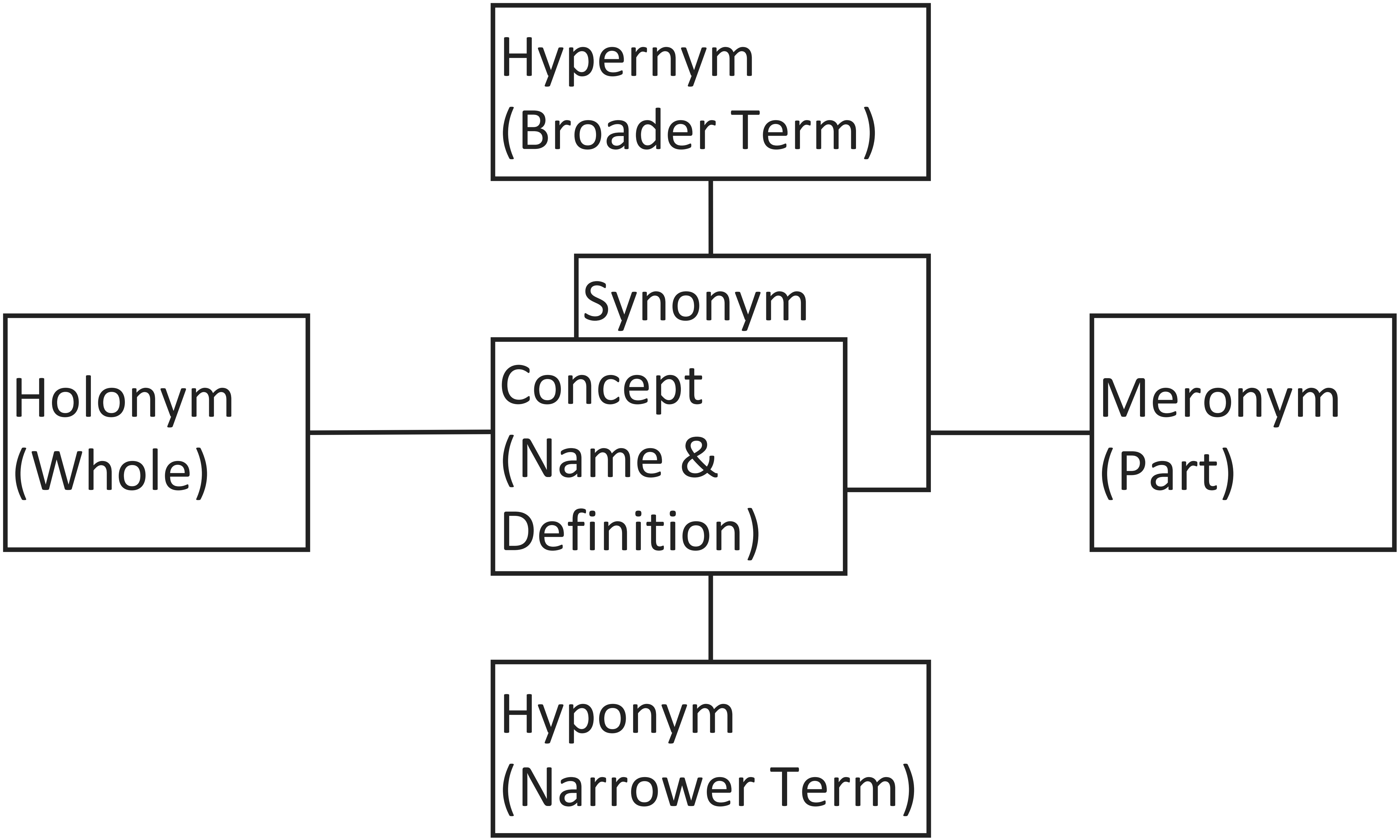
Syllabus

1	Introduction and Preprocessing	Text classification
2	Lexical semantics	Distributional semantics
3	Part of Speech Tagging	Probabilistic Sequence Modelling
4	Probabilistic Sequence Modelling	Context-Free Grammars
5	Probabilistic Parsing	Dependency parsing
	Easter holiday break	
6	N-gram language modelling	Deep learning for language models and tagging
7	Information Extraction	Question Answering
8	Topic Models	*ANZAC day holiday*
9	Information Retrieval -- Boolean search and the vector space model	Indexing and querying in the vector space model, evaluation
10	Index and vocabulary compression	Efficient query processing
11	The Web as a Graph: Page-rank & HITS	Machine Translation (word based)
12	Machine translation (phrase based) and neural encoder-decoder	Subject review

Outline

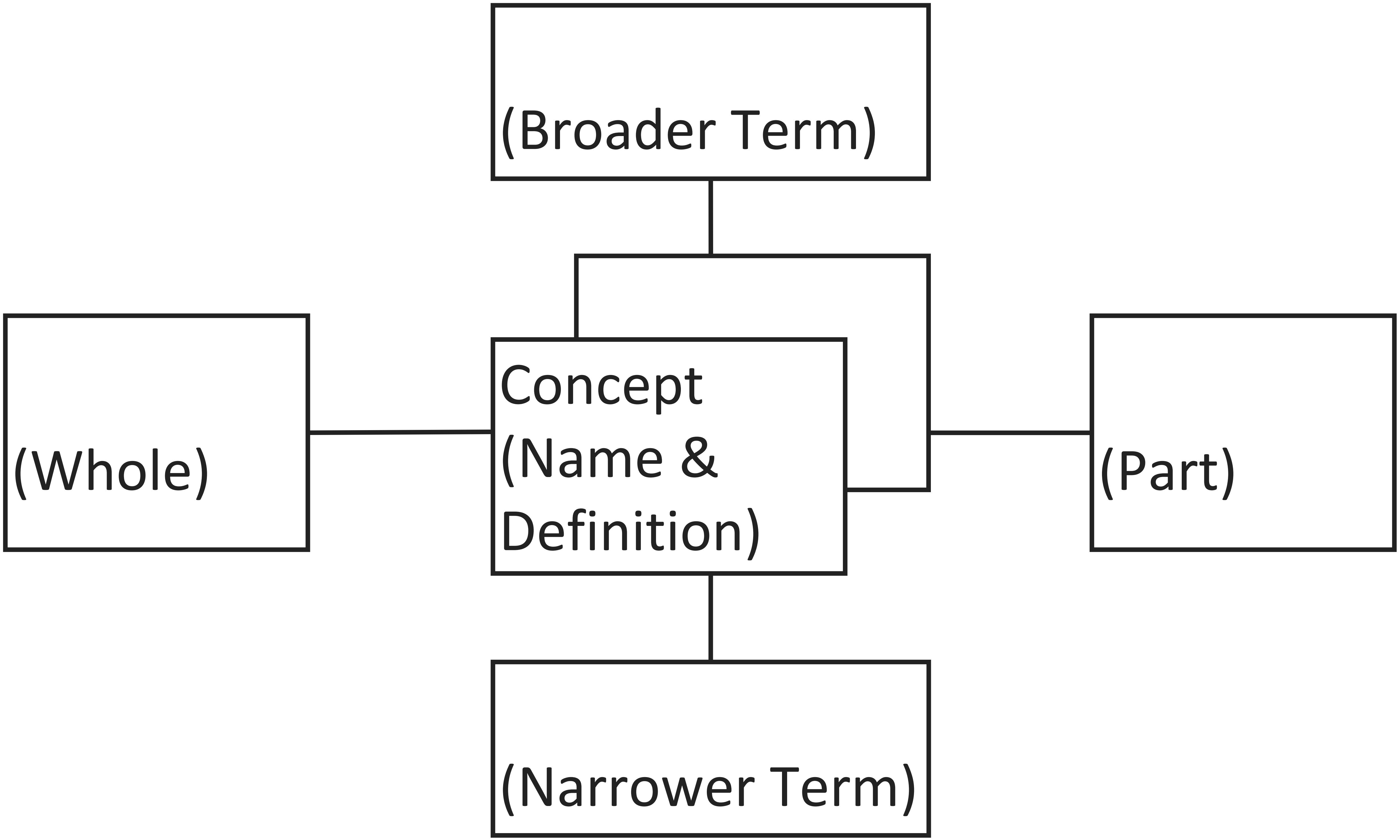
- Lexical semantics
  - Relationship between words
  - Wu & Palmer word similarity
- Distributional semantics (`WSTA_N4_distributional_semantics.ipynb`)
  - Point-wise Mutual Information (PMI)
  - Singular Value Decomposition (SVD)
  - Word embeddings

Relationship between words



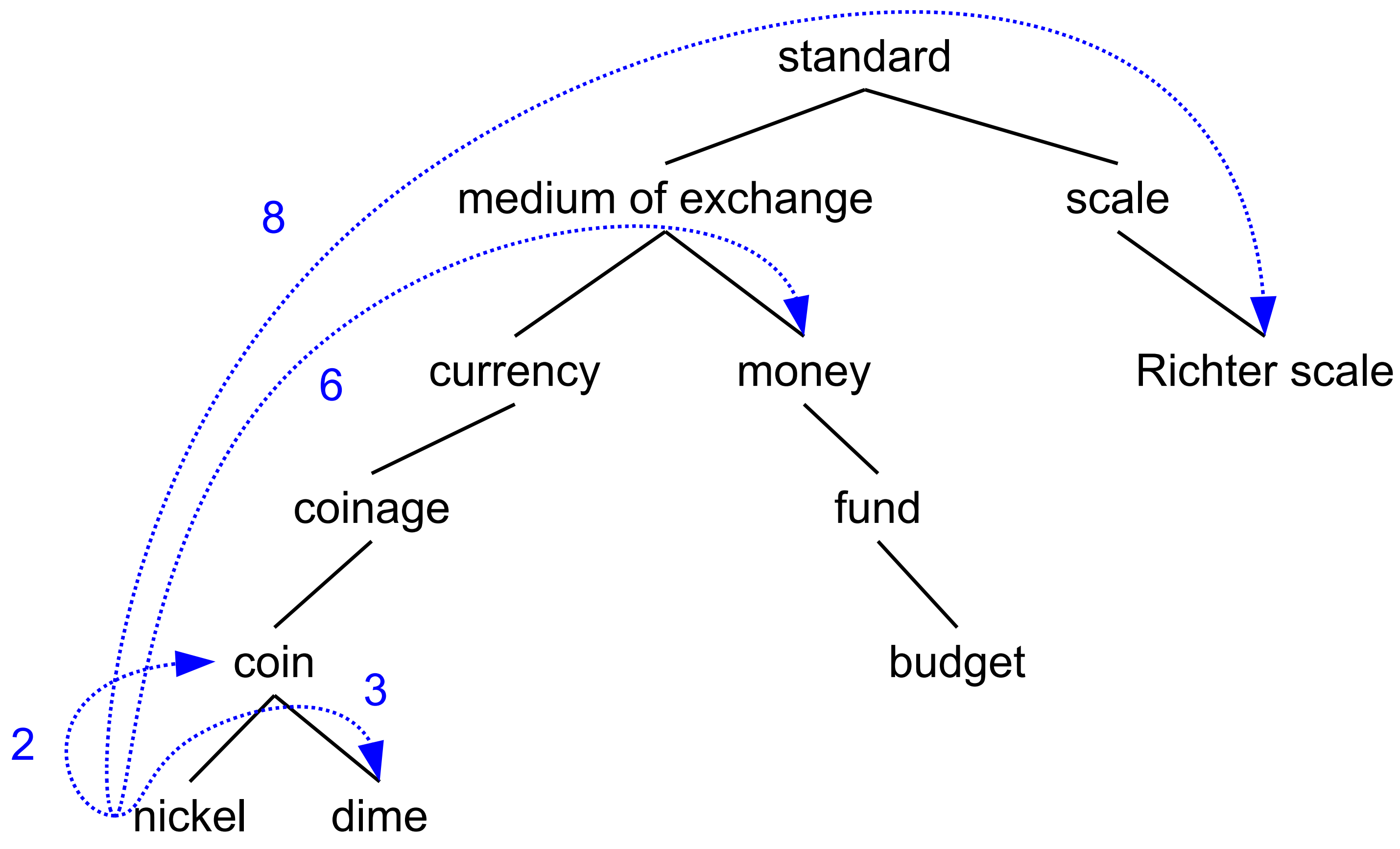
http://milyvicente2.blogspot.com.au/2013/03/the-nym-words.html

Relationship between words



http://milyvicente2.blogspot.com.au/2013/03/the-nym-words.html

Wu & Palmer word similarity

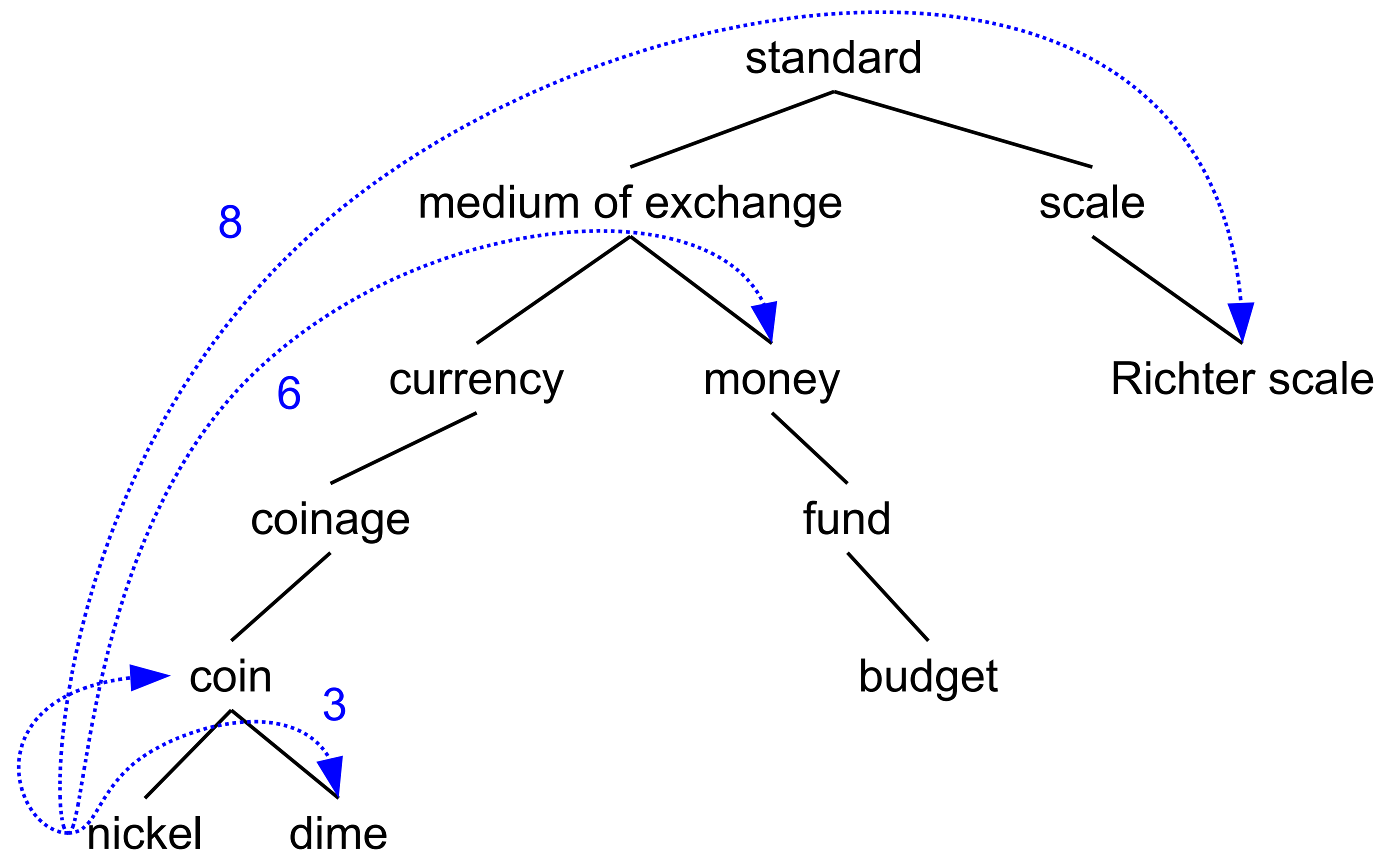


$$\text{simwup}(c_1,c_2) = \frac{2 * \text{depth}(\text{LCS}(c_1,c_2))}{\text{depth}(c_1) + \text{depth}(c_2)}$$

simwup(*nickel*,*money*) = 2*2/(3+ 6) = .44

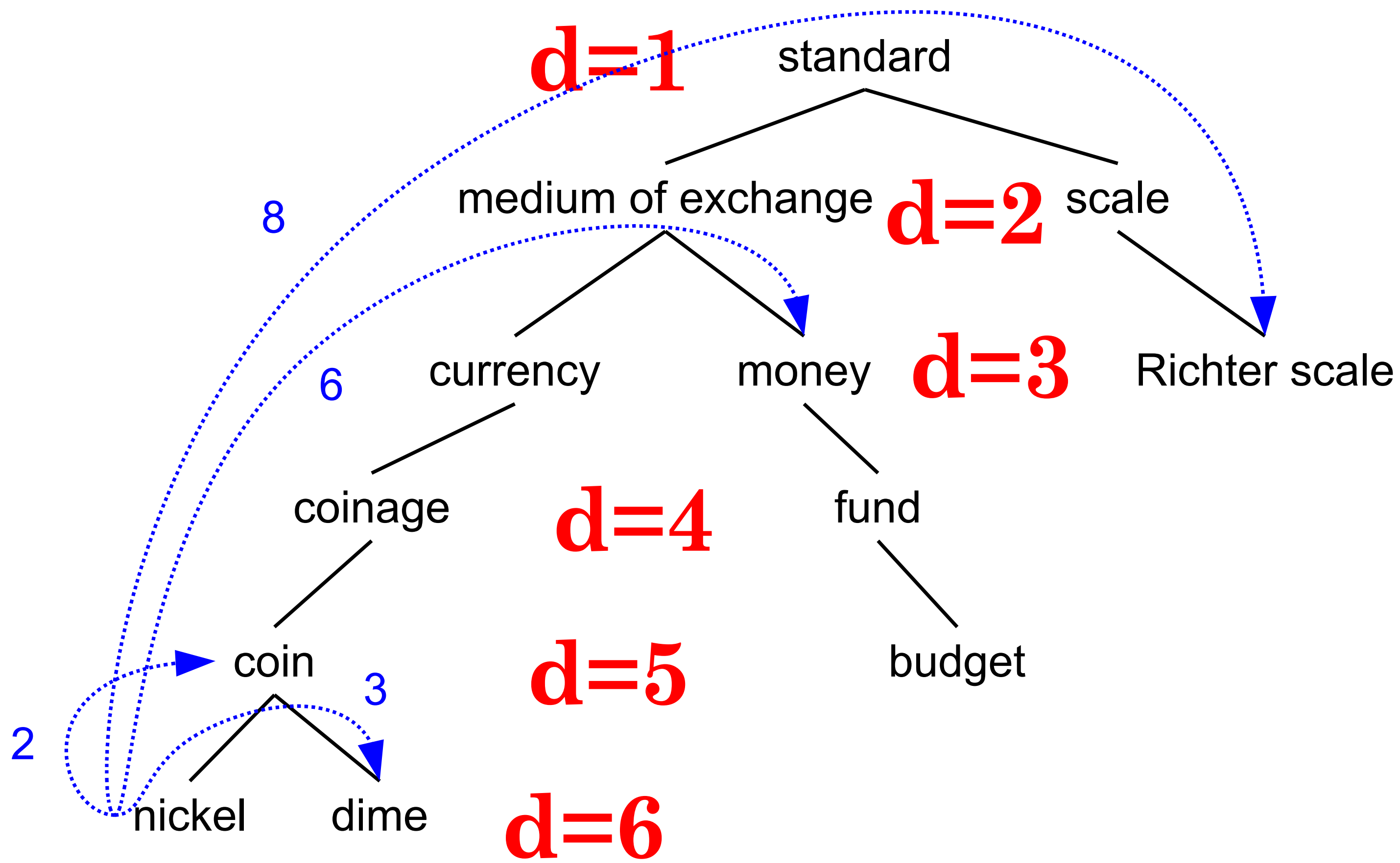
simwup(*nickel*,*Richter scale*) = 2*1/(3 + 6) = .22

LCS --- lowest common subsumer



- LCS(currency, money) = medium of exchange
- LCS(currency, scale) = standard
- LCS(nickel, money) = medium of exchange
- LCS(nickel, dime) = coin

Depth



$$\text{simwup}(c_1,c_2) = \frac{2*\text{depth}(\text{LCS}(c_1,c_2))}{\text{depth}(c_1) + \text{depth}(c_2)}$$

simwup(*nickel*,*money*) = 2*2/(3+ 6) = .44

simwup(*nickel*,*Richter scale*) = 2*1/(3 + 6) = .22

Outline

- Lexical semantics
  - Relationship between words
  - Wu & Palmer word similarity
- Distributional semantics (`WSTA_N4_distributional_semantics.ipynb`)
  - Point-wise Mutual Information (PMI)
  - Singular Value Decomposition (SVD)
  - Word embeddings

Pointwise mutual information

For two events x and y, pointwise mutual information (PMI) comparison between the actual joint probability of the two events (as seen in the data) with the expected probability under the assumption of independence

$$PMI(x,y) = \log_2 \frac{p(x,y)}{p(x)p(y)}$$

An example

- Corpus
- D1: a b c
- D2: a c d
- D3: b c d

- Context-word pairs:
- a-b, a-c, b-a, b-c, c-a, c-b
- a-c, a-d, c-a, c-d, d-a, d-c
- b-c, b-d, c-b, c-d, d-b, d-c

$$PMI(a,c) = \log_2 \frac{\frac{2}{18}}{\left(\frac{4}{18}\right) \times \left(\frac{6}{18}\right)}$$

	a	b	c	d	Σ
a		1	2	1	4
b	1		2	1	4
c	2	2		2	6
d	1	1	2		4
Σ	4	4	6	4	18

Another example

- Corpus
- D1: a b c
- D2: a c d
- D3: b c d

- Context-word pairs:
- a-b, b-a, b-c, c-b
- a-c, c-a, c-d, d-c
- b-c, c-b, c-d, d-c

$$PMI(a,c) = \log_2 \frac{\frac{1}{12}}{\left(\frac{2}{12}\right) \times \left(\frac{5}{12}\right)}$$

	a	b	c	d	Σ
a		1	1		2
b	1		2		3
c	1	2		2	5
d			2		2
Σ	2	3	5	2	12

CBOW & Skip-gram

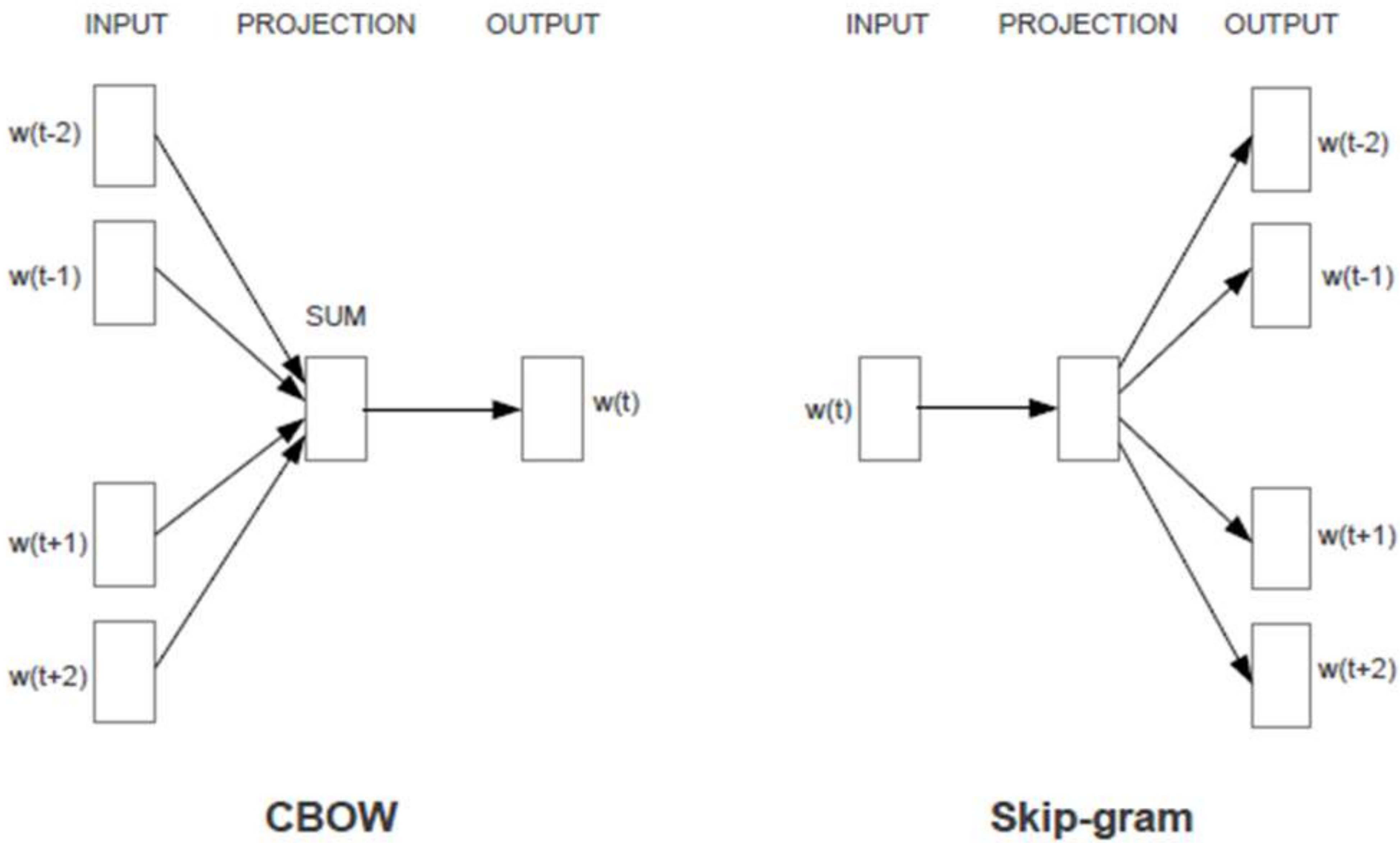
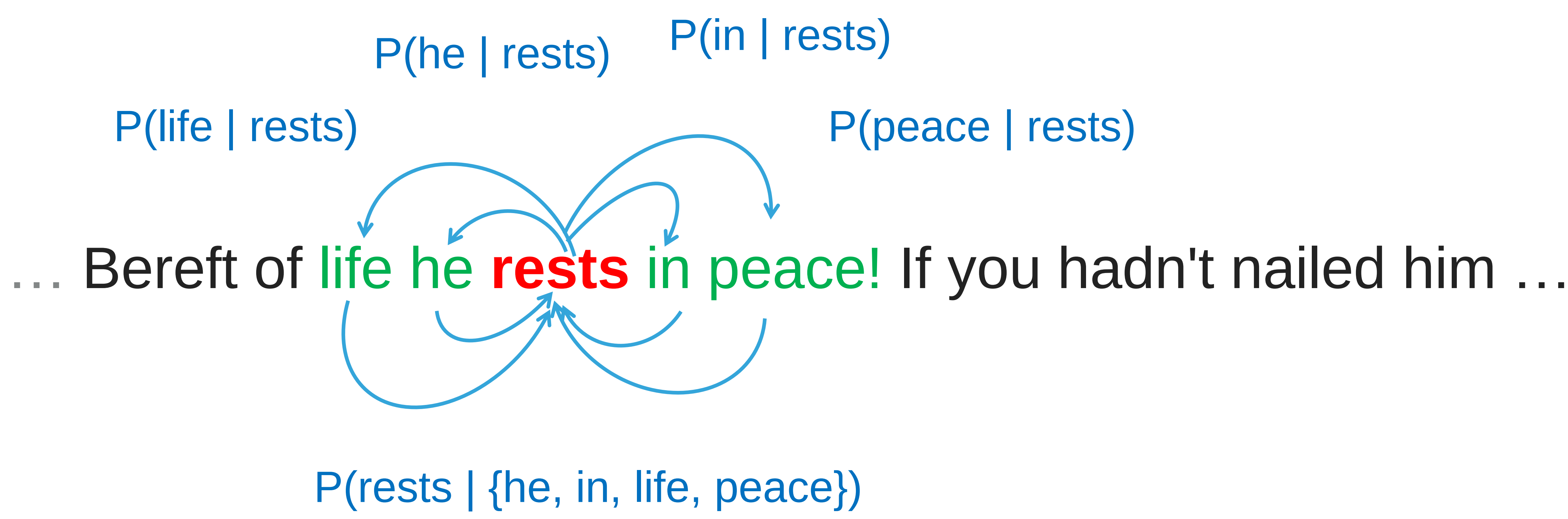


Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

Embeddings from predictions

- Framed as learning a classifier…
  - Skip-gram: predict words in local context surrounding given word



  - CBOW: predict word in centre, given words in the local surrounding context
- Local context means words within L positions, e.g., L=2