Word Sense Disambiguation

LING 571 — Deep Processing for NLP

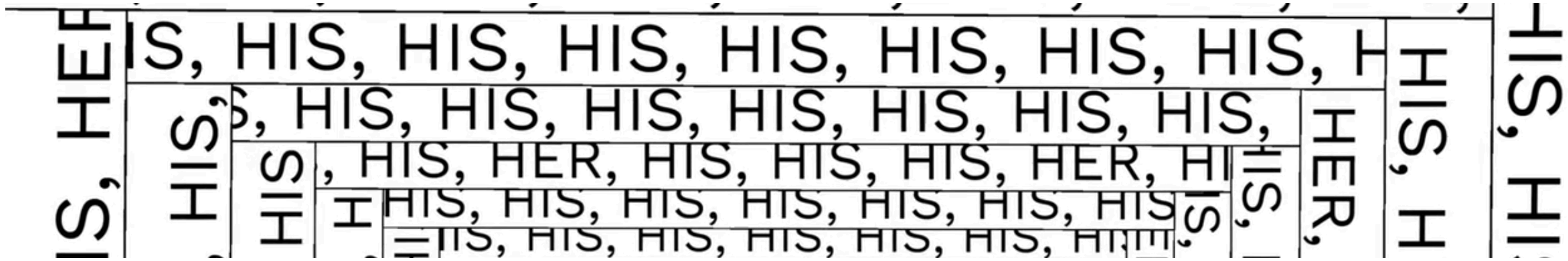
November 18, 2020

Shane Steinert-Threlkeld

In the News

A.I. Systems Echo Biases They're Fed, Putting Scientists on Guard

Researchers say computer systems are learning from lots and lots of digitized books and news articles that could bake old attitudes into new technology.



https://www.nytimes.com/2019/11/11/technology/artificial-intelligence-bias.html

[includes a quote from CLMS director/faculty Emily Bender]

Ambiguity of the Week

Lee Murray
@MurrayLeeA

syntax tree of the week

> **Model who burned down 3,500-year-old tree called 'The Senator' while high on meth avoids jail time**

Actually from 2014!

https://www.dailymail.co.uk/news/article-2652104/Model-burned-3-500-year-old-tree-called-The-Senator-high-meth-avoids-jail-time.html

Distributional Similarity for Word Sense Induction + Disambiguation

Word Sense Disambiguation

- We've looked at how to represent words
  - …so far, ignored **homographs**
- Wrong senses can lead to poor performance in downstream tasks
  - Machine translation, text classification
- Now, how do we go about differentiating homographs?

Word Senses

WordNet Sense	Spanish Translation	Roget Category	Word in Context
bass[4]	*lubina*	FISH/INSECT	…fish as Pacific salmon and striped **bass** and…
bass[4]	*lubina*	FISH/INSECT	…produce filets of smoked **bass** or sturgeon…
bass[7]	*bajo*	MUSIC	…exciting jazz **bass** player since Ray Brown…
bass[7]	*bajo*	MUSIC	…play **bass** because he doesn't have to solo…

WSD With Distributional Similarity

- We’ve covered how to create vectors for *words*, but how do we represent *senses*?

WSD With Distributional Similarity

- We've covered how to create vectors for *words*, but how do we represent *senses*?

- ***First order*** vectors:
  - $\vec{w} = (f_1, f_2, f_3 \ldots)$
  - Feature vector of word itself

WSD With Distributional Similarity

- We've covered how to create vectors for *words*, but how do we represent *senses*?

- ***First order*** vectors:
  - $\vec{w} = (f_1, f_2, f_3 \ldots)$
  - Feature vector of word itself

- ***Second order*** vectors:
  - Context vector

Word Representation

- 2nd Order Representation:
- Identify words in context of w
- For each x in context of w:
  - Compute x vector representation
- Compute centroid of these $\vec{x}$ vector representations

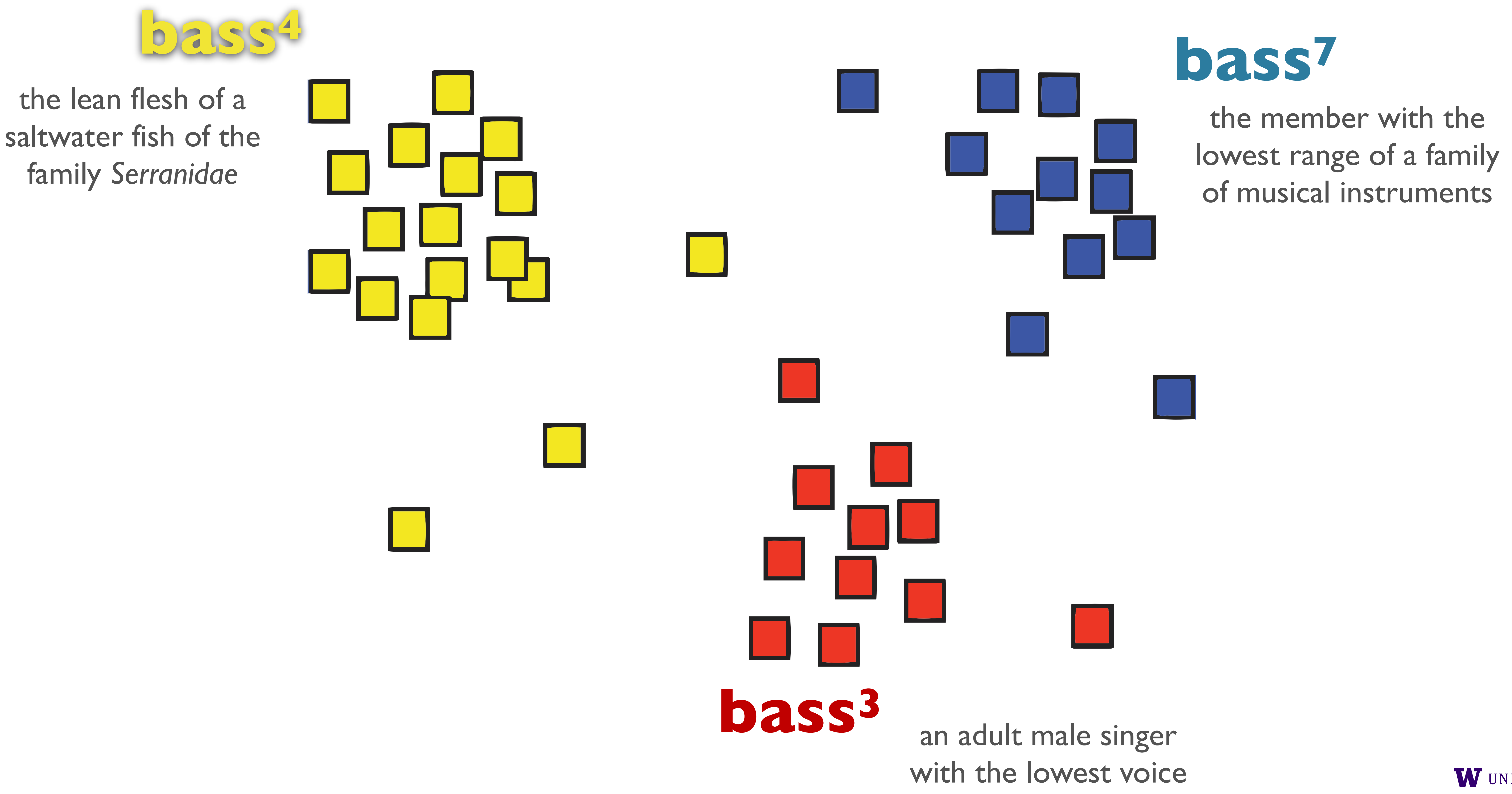
Computing Word Senses

- Compute context vector for each occurrence of word in corpus
- Cluster these context vectors
  - # of clusters = # of senses
- Cluster centroid represents word sense
- Link to specific sense?
  - Pure unsupervised: no sense tag, just i^{th} sense
  - Some supervision: hand label clusters, or tag training

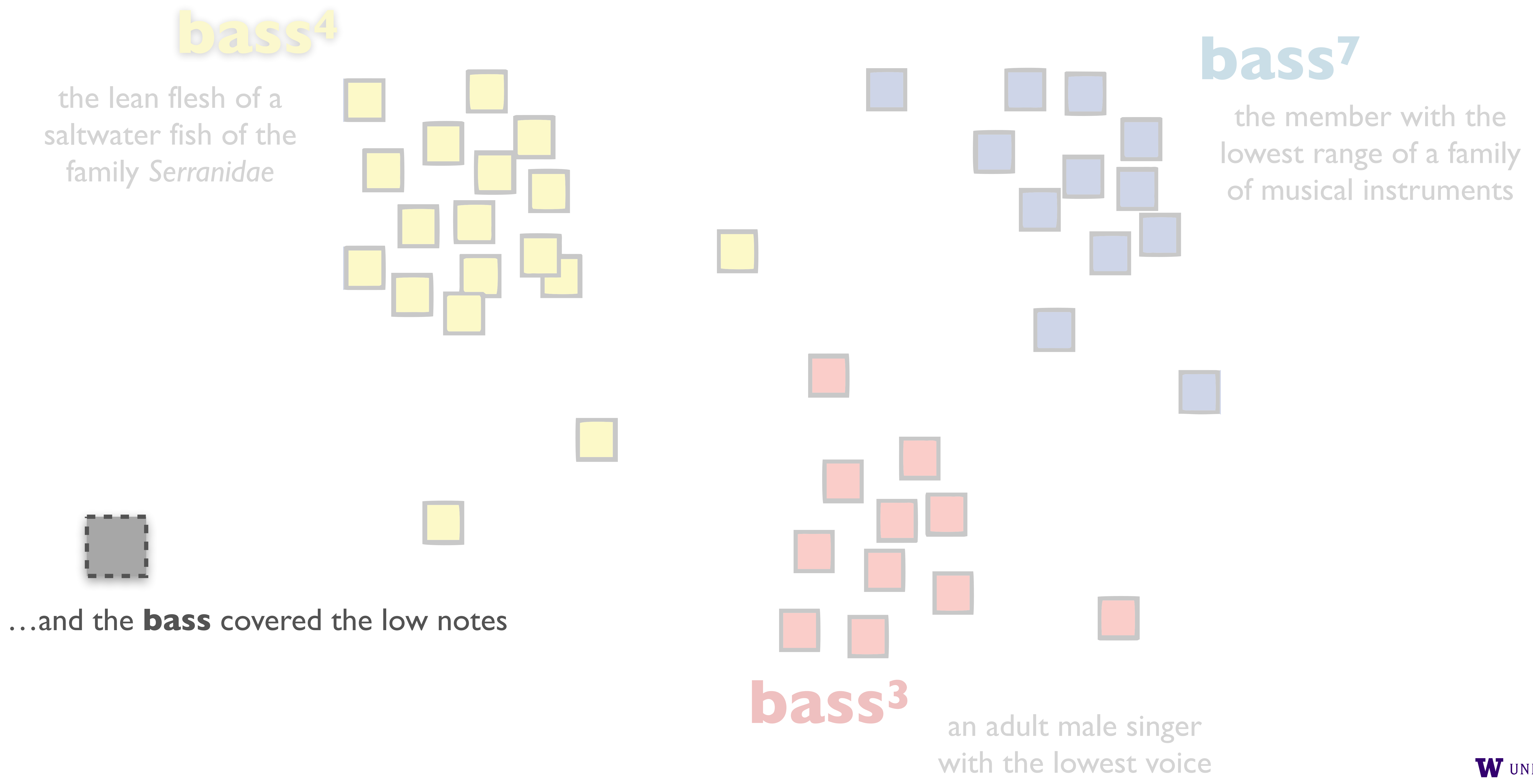
Disambiguating Instances

- To disambiguate an instance ***t*** of ***w***:
  - Compute context vector for instance
  - Retrieve all senses of ***w***
  - Assign ***w*** sense with closest centroid to ***t***

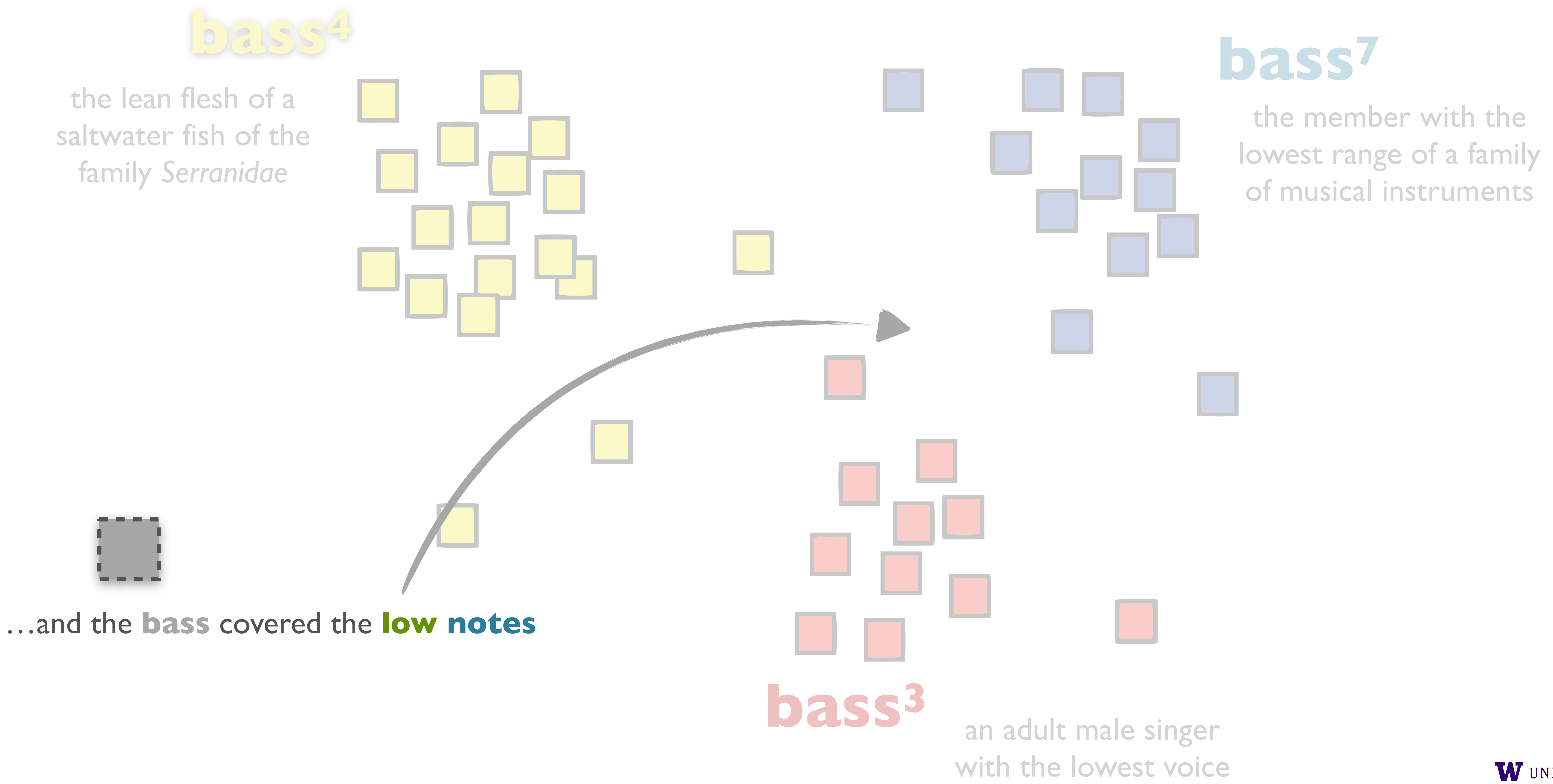
Computing Word Senses



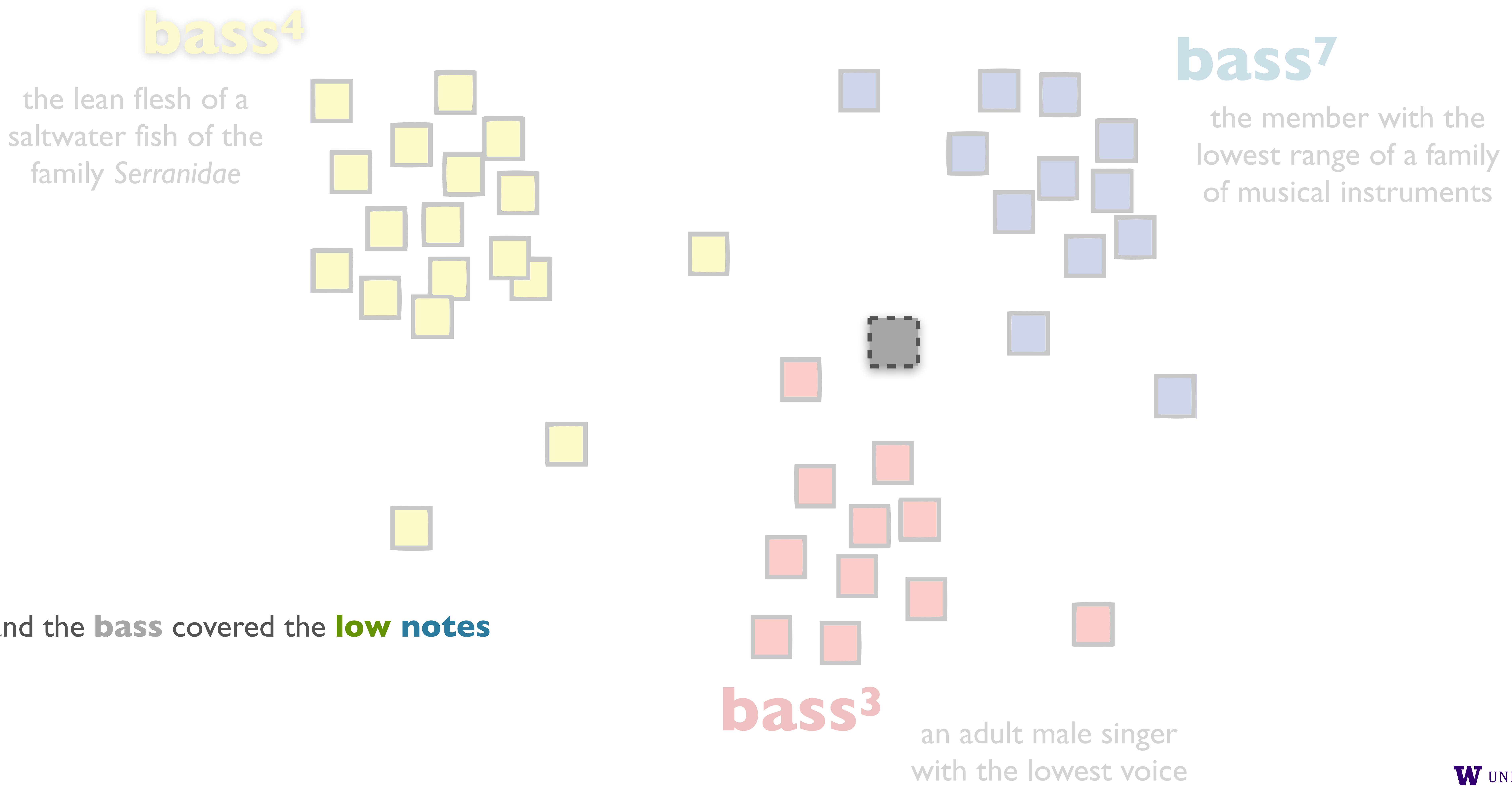
Computing Word Senses



Computing Word Senses

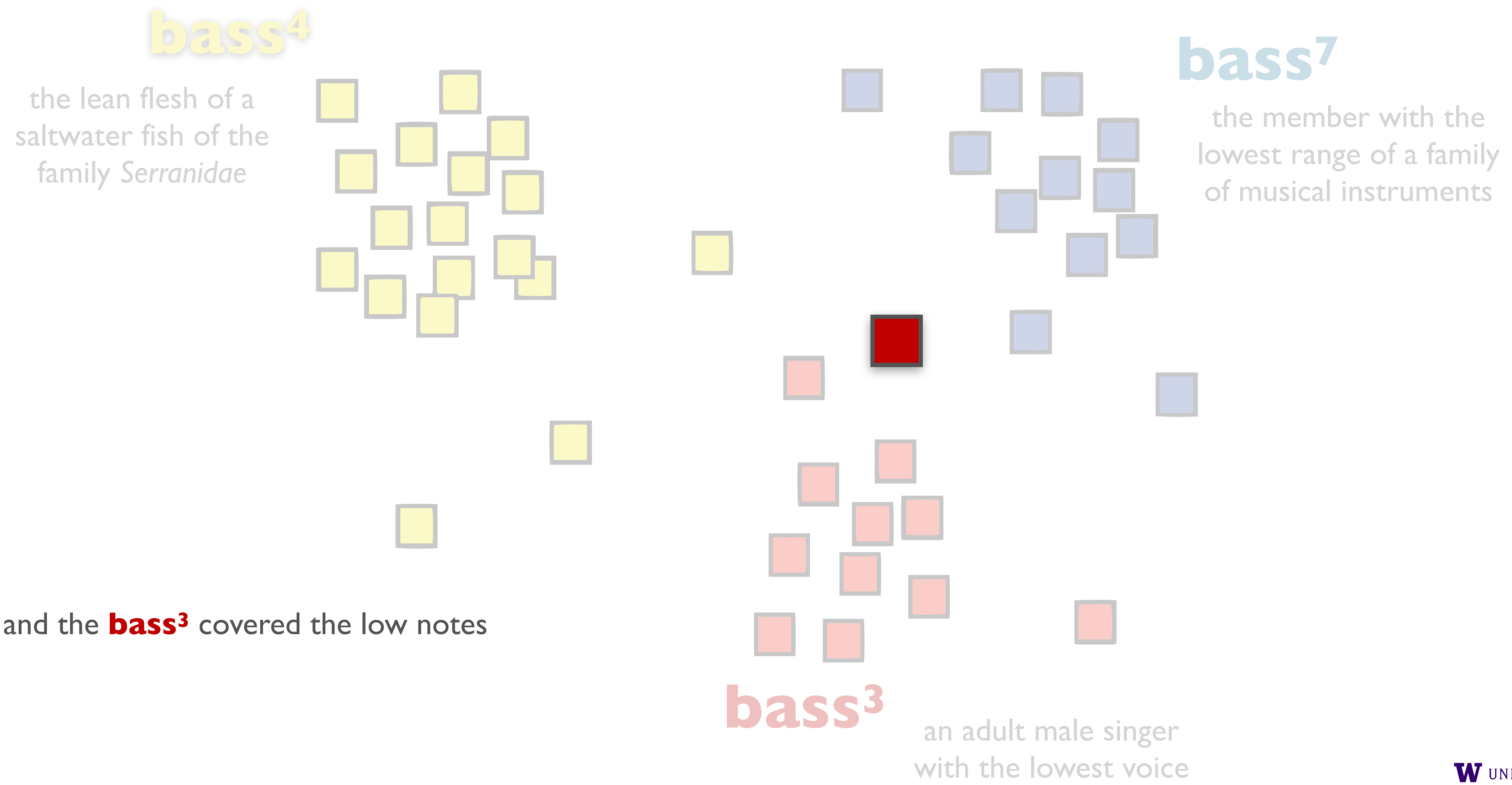


Computing Word Senses



…and the **bass** covered the **low notes**

Computing Word Senses



…and the **bass³** covered the low notes

Local Context Clustering

- "Brown" (aka IBM) clustering [link]
  - **Generative**, **class-based** language model over adjacent words
    - class-based:
      - Each w_i has class c_i
      - The distribution for words given a class: $P(w|c)$
    - Generative:
      - Can estimate the probability of the current set of senses in the corpus, given the current set of clusters:

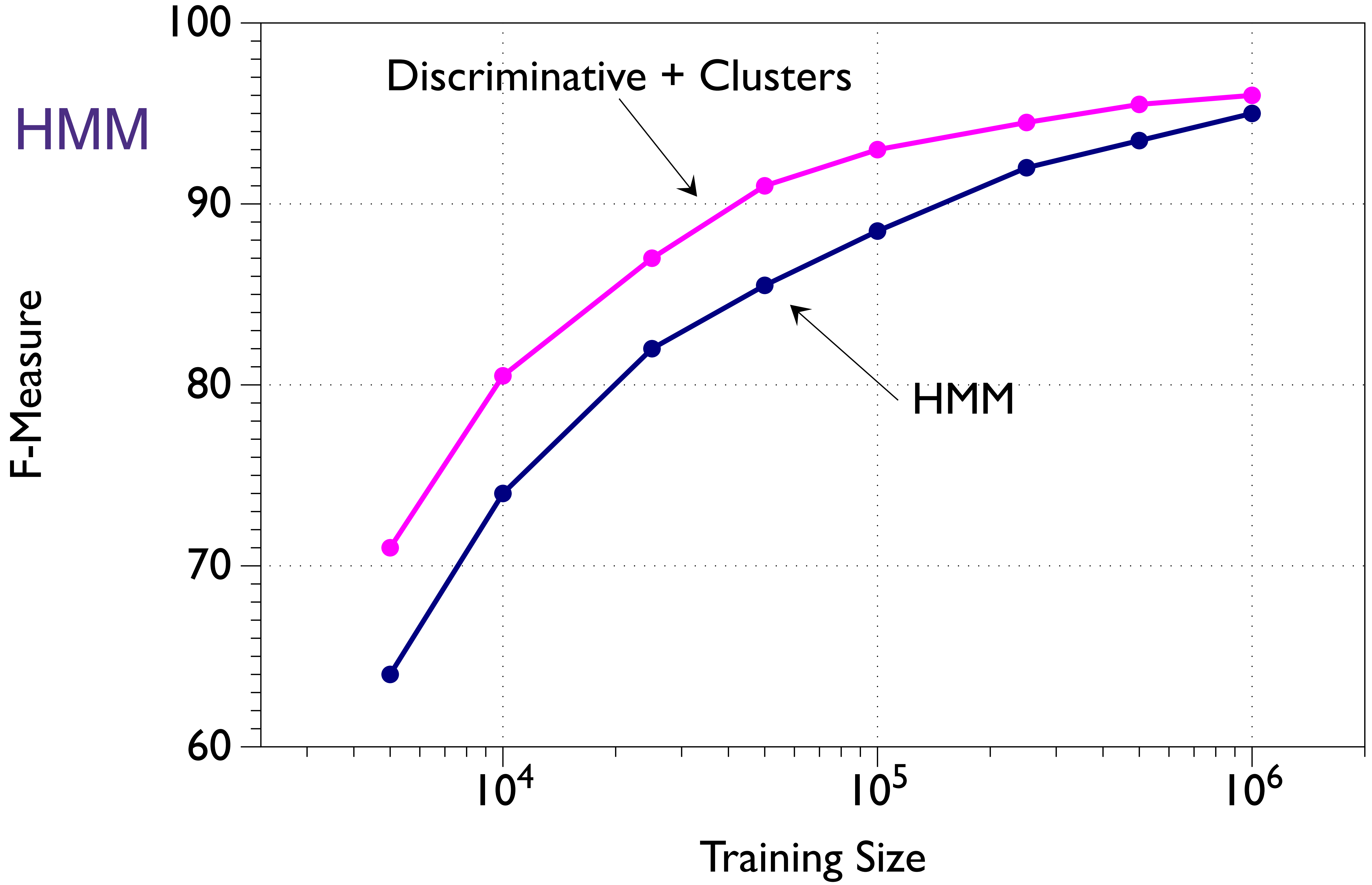
$$\log P(corpus \mid C) = \sum_i \log P(w_i \mid c_i) + \log P(c_i \mid c_{i-1})$$

Local Context Clustering

- Greedy, hierarchical clustering $\log P(corpus \mid C) = \sum_i \log P(w_i \mid c_i) + \log P(c_i \mid c_{i-1})$
  1. Start with each word in own cluster
  2. Merge clusters which decrease the likelihood the least — maximize $P(corpus)$
  3. Proceed until all words in one cluster

Clustering Impact

- Improves downstream tasks
  - Named Entity Recognition vs. HMM
    - Miller et al '04



Contextual Embeddings for Disambiguation

Average of all contextual embeddings from dataset with a given sense label
[in principle, could be centroid of cluster]

Nearest neighbor classification

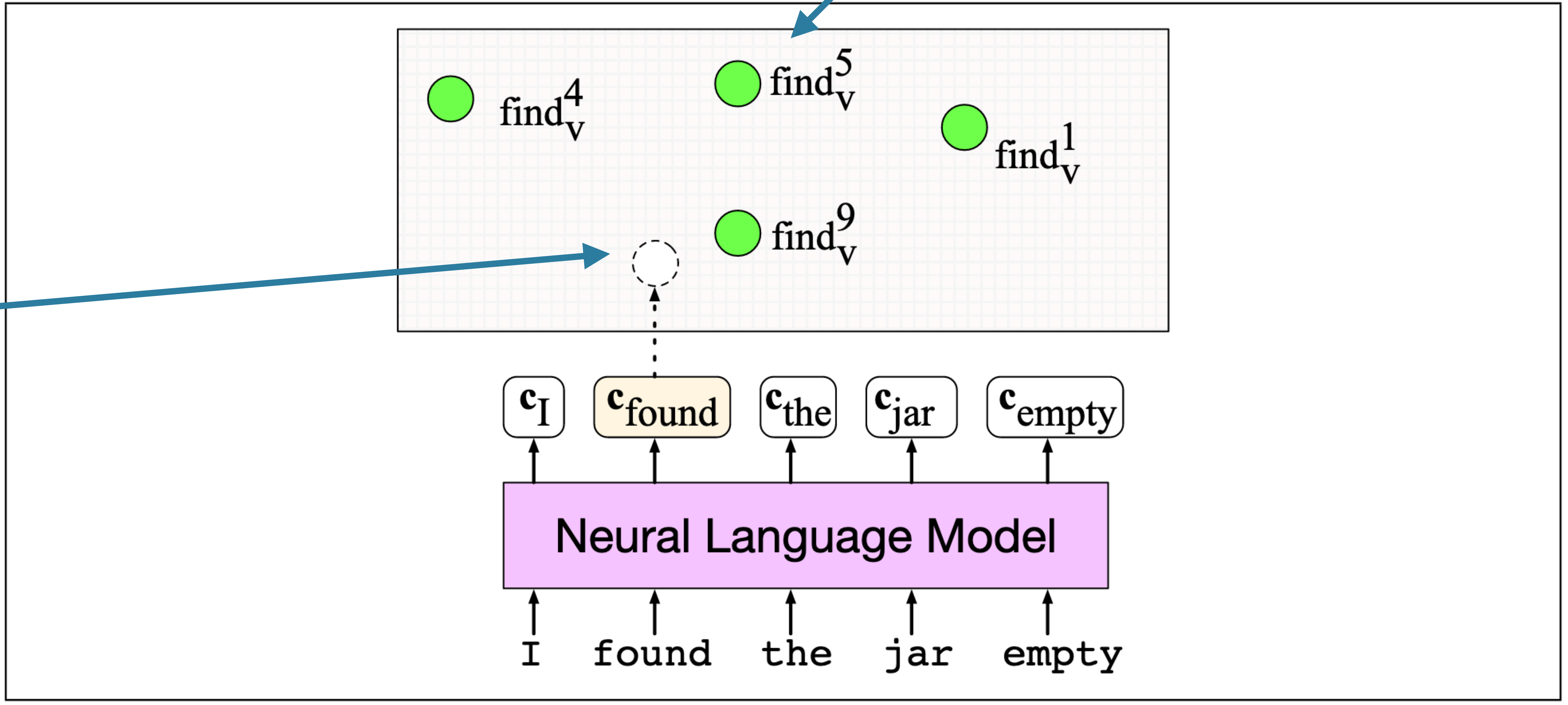


Figure 19.9 The nearest-neighbor algorithm for WSD. In green are the contextual embeddings precomputed for each sense of each word; here we just show a few of the senses for *find*. A contextual embedding is computed for the target word *found*, and the and then the nearest neighbor sense (in this case $\mathbf{find}_n^9$) would be chosen. Figure inspired by Loureiro and Jorge (2019).

Resource-Based Models

Resource-Based Models

- Alternative to just clustering distributional representations
- What if we actually have some resources?
  - Dictionaries
  - Semantic sense taxonomy
  - Thesauri

Dictionary-Based Approach

- (Simplified) Lesk algorithm
  - "How to tell a pine cone from an ice cream cone" (Lesk, 1986)
- Compute "signature" of word senses:
  - Words in gloss and examples in dictionary

bank (n.)	1	a financial institution that accepts deposits and channels the money into lending activities. *"he cashed a check at the bank,""that bank holds the mortgage on my home."*
	2	sloping land (especially the slope beside a body of water). *"they pulled the canoe up on the bank,""he sat on the bank of the river and watched the currents."*

Dictionary-Based Approach

- Compute context of word to disambiguate
- Compare overlap between signature and context
- Select sense with highest (non-stopword) overlap

bank (n.)	1	a financial institution that accepts deposits and channels the money into lending activities. *"he cashed a check at the bank," "that bank holds the mortgage on my home."*
	2	sloping land (especially the slope beside a body of water). *"they pulled the canoe up on the bank," "he sat on the bank of the river and watched the currents."*

Dictionary-Based Approach

- Compute context of word to disambiguate
- Compare overlap between signature and context
- Select sense with highest (non-stopword) overlap

*"She went to the **bank** to withdraw some money."*

bank (n.)	1	a financial institution that accepts deposits and channels the money into lending activities. *"he cashed a check at the bank," "that bank holds the mortgage on my home."*
	2	sloping land (especially the slope beside a body of water). *"they pulled the canoe up on the bank," "he sat on the bank of the river and watched the currents."*

Dictionary-Based Approach

- Compute context of word to disambiguate
- Compare overlap between signature and context
- Select sense with highest (non-stopword) overlap

*"She went to the **bank** to withdraw some money."*

bank (n.)	1	a financial institution that accepts deposits and channels the money into lending activities. *"he cashed a check at the bank," "that bank holds the mortgage on my home."*
	2	sloping land (especially the slope beside a body of water). *"they pulled the canoe up on the bank," "he sat on the bank of the river and watched the currents."*

Dictionary-Based Approach

- Compute context of word to disambiguate
- Compare overlap between signature and context
- Select sense with highest (non-stopword) overlap

"The frog sat on the river ***bank****, half in and half out of the water."*

bank (n.)	1	a financial institution that accepts deposits and channels the money into lending activities. *"he cashed a check at the bank," "that bank holds the mortgage on my home."*
	2	sloping land (especially the slope beside a body of water). *"they pulled the canoe up on the bank," "he sat on the bank of the river and watched the currents."*

Dictionary-Based Approach

- Compute context of word to disambiguate
- Compare overlap between signature and context
- Select sense with highest (non-stopword) overlap

"The frog sat on the river ***bank****, half in and half out of the water."*

bank (n.)	1	a financial institution that accepts deposits and channels the money into lending activities. *"he cashed a check at the bank," "that bank holds the mortgage on my home."*
	2	sloping land (especially the slope beside a body of water). *"they pulled the canoe up on the bank," "he sat on the bank of the river and watched the currents."*

Sense Taxonomy/Thesaurus Approaches

WordNet Taxonomy

- Widely-used English sense resource

WordNet Taxonomy

- Widely-used English sense resource
- Manually constructed lexical database

WordNet Taxonomy

- Widely-used English sense resource
- Manually constructed lexical database
  - 3 tree-structured hierarchies
    - Nouns (117K)
    - Verbs (11K)
    - Adjective+Adverb (27K)

WordNet Taxonomy

- Widely-used English sense resource
- Manually constructed lexical database
  - 3 tree-structured hierarchies
    - Nouns (117K)
    - Verbs (11K)
    - Adjective+Adverb (27K)
  - Entries:
    - Synonym set (“***synset***”)
    - Gloss
    - Example usage

WordNet Taxonomy

- Relations between entries:
  - Synonymy: in synset
  - Hyponym/Hypernym: *is-a* tree

WordNet

The **noun** “bass” has **8** senses in WordNet. [link]

1. **bass**^{1} - (the lowest part of the musical range)
2. **bass**^{2}, **bass part**^{1} - (the lowest part in polyphonic music)
3. **bass**^{3}, **basso**^{1} - (an adult male singer with the lowest voice)
4. **sea bass**^{1}, **bass**^{4} - (the lean fish of a saltwater fish of the family *Serranidae*)
5. **freshwater bass**^{1}, **bass**^{5} - (any of various North American freshwater fish with lean flesh (especially of the genus *Micropterus*))
6. **bass**^{6}, **bass voice**^{1}, **basso**^{2} - (the lowest adult male singing voice)
7. **bass**^{7} - (the member with the lowest range of a family of musical instruments)
8. **bass**^{8} - (nontechnical name for any numerous edible marine and freshwater spiny-finned fishes)

The **adjective** “bass” has **1** sense in WordNet.

1. **bass**^{1} - deep6 - (having or denoting a low vocal or instrumental range)
   “a deep voice”;”a bass voice is lower than a baritone voice”;”a bass clarinet”

Noun WordNet Relations

Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	*breakfast*^{1} → *meal*^{1}
Hyponym	Subordinate	From concepts to subtypes	*meal*^{1} → *lunch*^{1}
Instance Hypernym	Instance	From instances to their concepts	*Austen*^{1} → *author*^{1}
Instance Hyponym	Has-Instance	From concepts to concept instances	*composer*^{1} → *Bach*^{1}
Member Meronym	Has-Member	From groups to their members	*faculty*^{2} → *professor*^{1}
Member Holonym	Has-Part	From members to their groups	*copilot*^{1} → *crew*^{1}
Part Meronym	Part-Of	From wholes to parts	*table*^{2} → *leg*^{3}
Part Holonym		From parts to wholes	*course*^{7} → *meal*^{1}
Substance Meronym		From substances to their subparts	*water*^{1} → *oxygen*^{1}
Substance Holonym		From parts of substances to wholes	*gin*^{1} → *martini*^{1}
Antonym		Semantic opposition between lemmas	*leader*^{1} ⟺ *follower*^{1}
Derivationally Related Form		Lemmas	*destruction*^{1} ⟺ *destroy*^{1}

WordNet Taxonomy

```
Sense 3
bass, basso --
(an adult male singer with the lowest voice)
  => singer, vocalist, vocalizer, vocaliser
    => musician, instrumentalist, player
      => performer, performing artist
        => entertainer
          => person, individual, someone…
            => organism, being
              => living thing, animate thing
                => whole, unit
                  => object, physical object
                    => physical entity
                      => entity
            => causal agent, cause, causal agency
              => physical entity
                => entity
```

Thesaurus-based Techniques

- Key idea:
  - The number of “hops” between words in a thesaurus can be a distance measure

Thesaurus-based Techniques

- Key idea:
  - The number of “hops” between words in a thesaurus can be a distance measure
  - The shorter path length in thesaurus, smaller semantic distance

Thesaurus-based Techniques

- Key idea:
  - The number of “hops” between words in a thesaurus can be a distance measure
  - The shorter path length in thesaurus, smaller semantic distance
  - Words similar to parents, siblings in tree

Thesaurus-based Techniques

- Key idea:
  - The number of “hops” between words in a thesaurus can be a distance measure
  - The shorter path length in thesaurus, smaller semantic distance
  - Words similar to parents, siblings in tree
- `pathlength` = #edges in shortest route through graph between nodes
  - $sim_{path} = -\log \ pathlen(c_1,\ c_2)$ [Leacock & Chodorow, 1998]

Problem #1

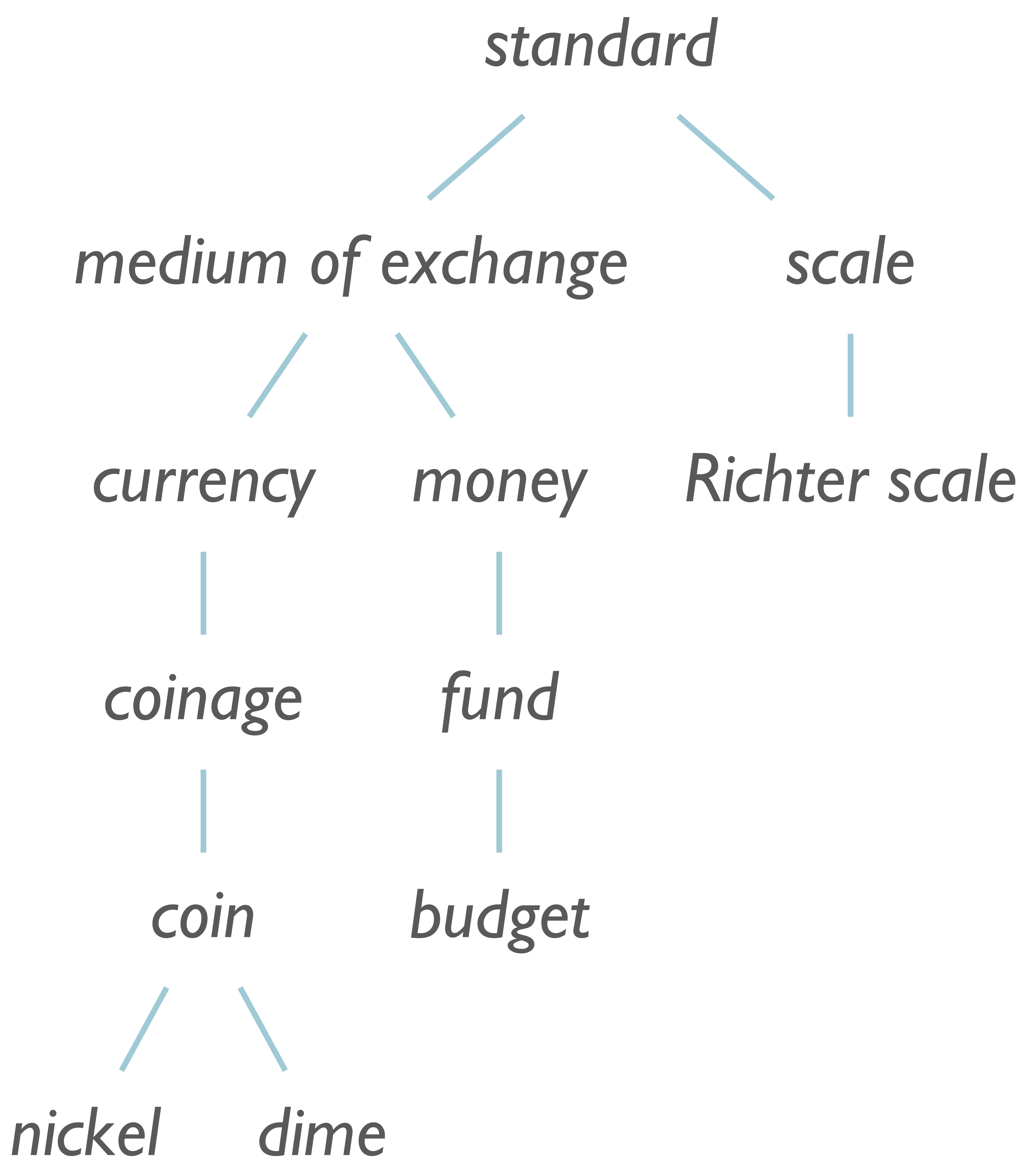
- Rarely know ***which sense***, thus rarely know ***which node***

Problem #1

- Rarely know ***which sense***, thus rarely know ***which node***
- **Solution**
  - assume most similar senses as an estimate
  - $wordsim(w_1, w_2) = \max sim(c_1, c_2)$

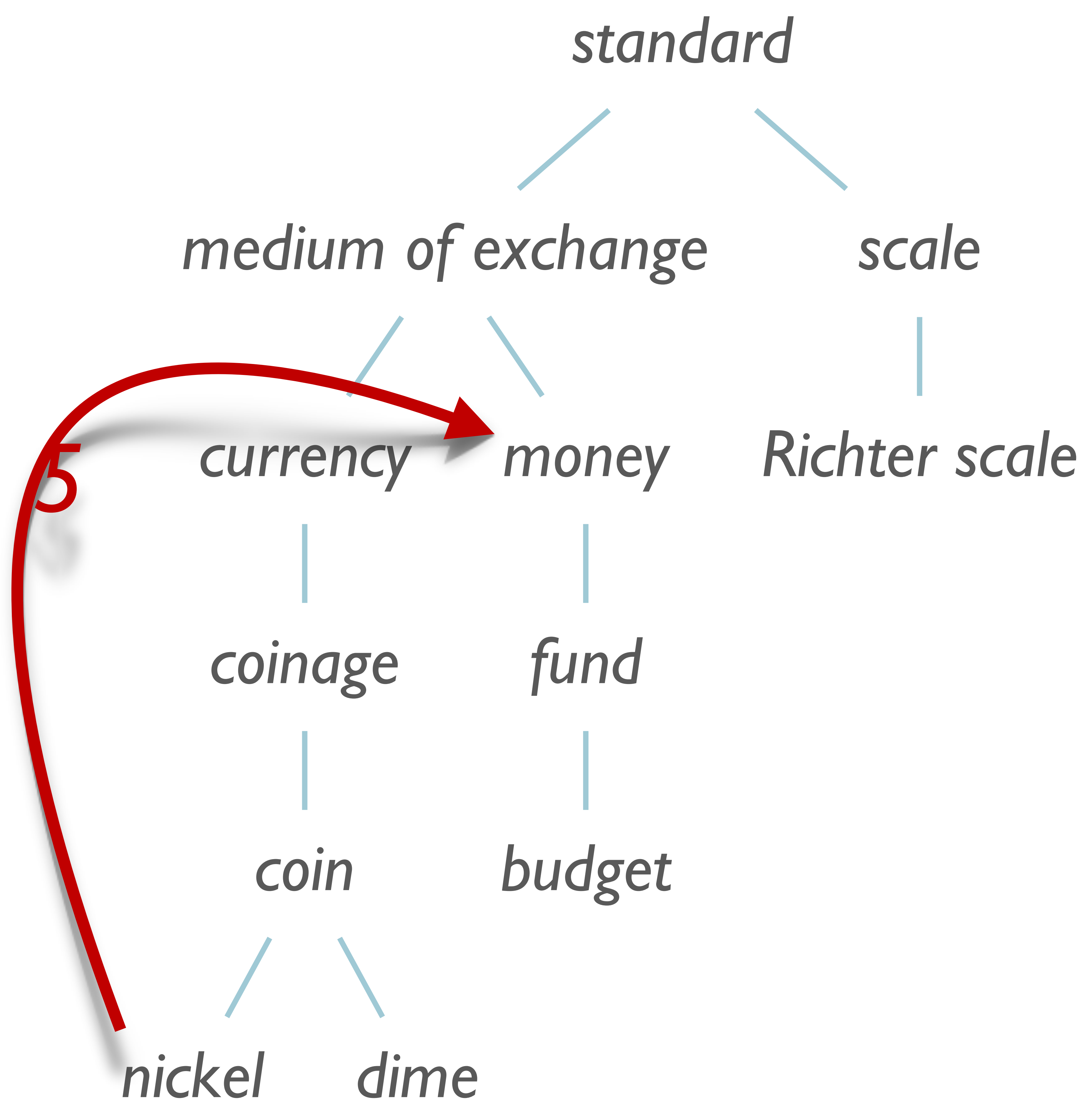
Problem #2

- Links in WordNet not uniformly different



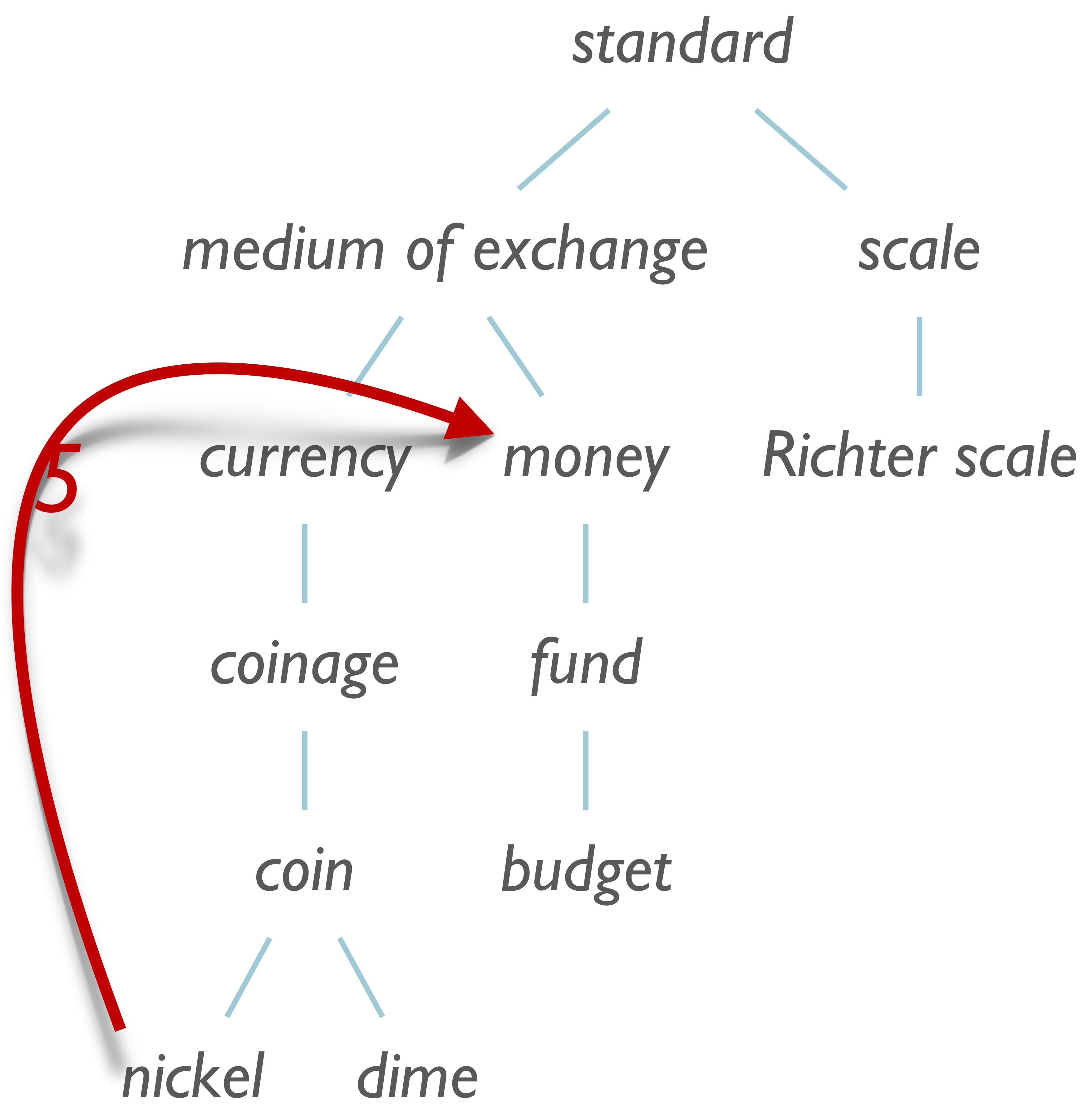
Problem #2

- Links in WordNet not uniformly different
  - |Nickel → Money| = 5



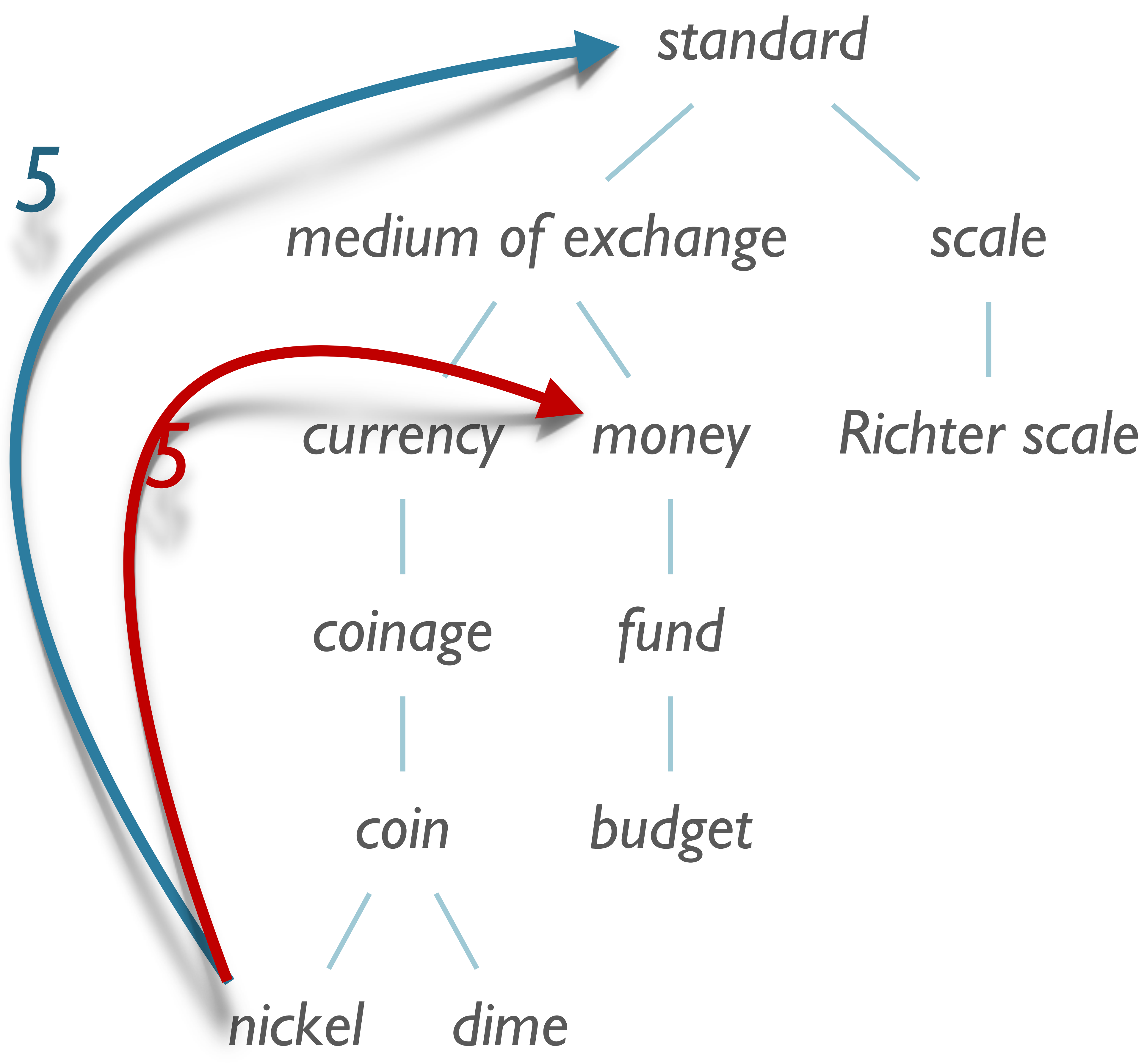
Problem #2

- Links in WordNet not uniformly different
  - lNickel → Moneyl = 5
  - lNickel → Standardl = 5



Problem #2

- Links in WordNet not uniformly different
  - |Nickel → Money| = 5
  - |Nickel → Standard| = 5
- How to capture?



Thesaurus-based Techniques: A Solution

- Add ***information content*** from a corpus (Resnik, 1995)
- $P(c)$: probability that a word is instance of concept c
- $words(c)$: words subsumed by concept c;
- N: words in corpus

$$P(c) = \frac{\sum_{w \in words(c)} count(w)}{N}$$

Information Content

- Using a sense-tagged corpus (like SemCor)

```
…
<wf cmd="ignore" pos="IN">in</wf>
<wf cmd="ignore" pos="DT">the</wf>
<wf cmd="done" pos="NN" lemma="united_states_of_america" wnsn="1" lexsn="1:15:00::">United_States_of_America</wf>
<wf cmd="done" pos="VB" lemma="be" wnsn="1" lexsn="2:42:03::">was</wf>
<wf cmd="done" pos="JJ" lemma="gay" wnsn="6" lexsn="5:00:00:homosexual:00">gay</wf>
<punc>,</punc>
<wf cmd="done" pos="JJ" lemma="witty" wnsn="1" lexsn="5:00:00:humorous:00">witty</wf>
<punc>,</punc>
<wf cmd="done" pos="JJ" lemma="mercurial" wnsn="1" lexsn="5:00:00:changeable:00">mercurial</wf>
<punc>,</punc>
<wf cmd="done" pos="JJ" lemma="full" wnsn="1" lexsn="3:00:00::">full</wf>
<wf cmd="done" pos="JJ" ot="notag">of</wf>
<wf cmd="done" pos="NN" lemma="prank" wnsn="1" lexsn="1:04:01::">pranks</wf>
<wf cmd="ignore" pos="CC">and</wf>
<wf cmd="done" pos="NN" ot=“foreignword">bonheur</wf>
…
```

“The Serge Prokofieff whom we knew in the United States of America was gay, witty, mercurial, full of pranks and bonheur–

Concept Probability Example

entity
0.395

↑

inanimate-object
0.167

↑

natural-object
0.0163

↑

geological-formation
0.00176

↗ ↖

natural-elevation 0.000113 | shore 0.0000836

↑ ↑

hill 0.0000189 | coast 0.0000216

Information Content-Based Similarity Measures

- Information content of node (concept c)
  - $IC(c) = -\log P(c)$
  - As probability of encountering c increases, informativeness decreases

Information Content-Based Similarity Measures

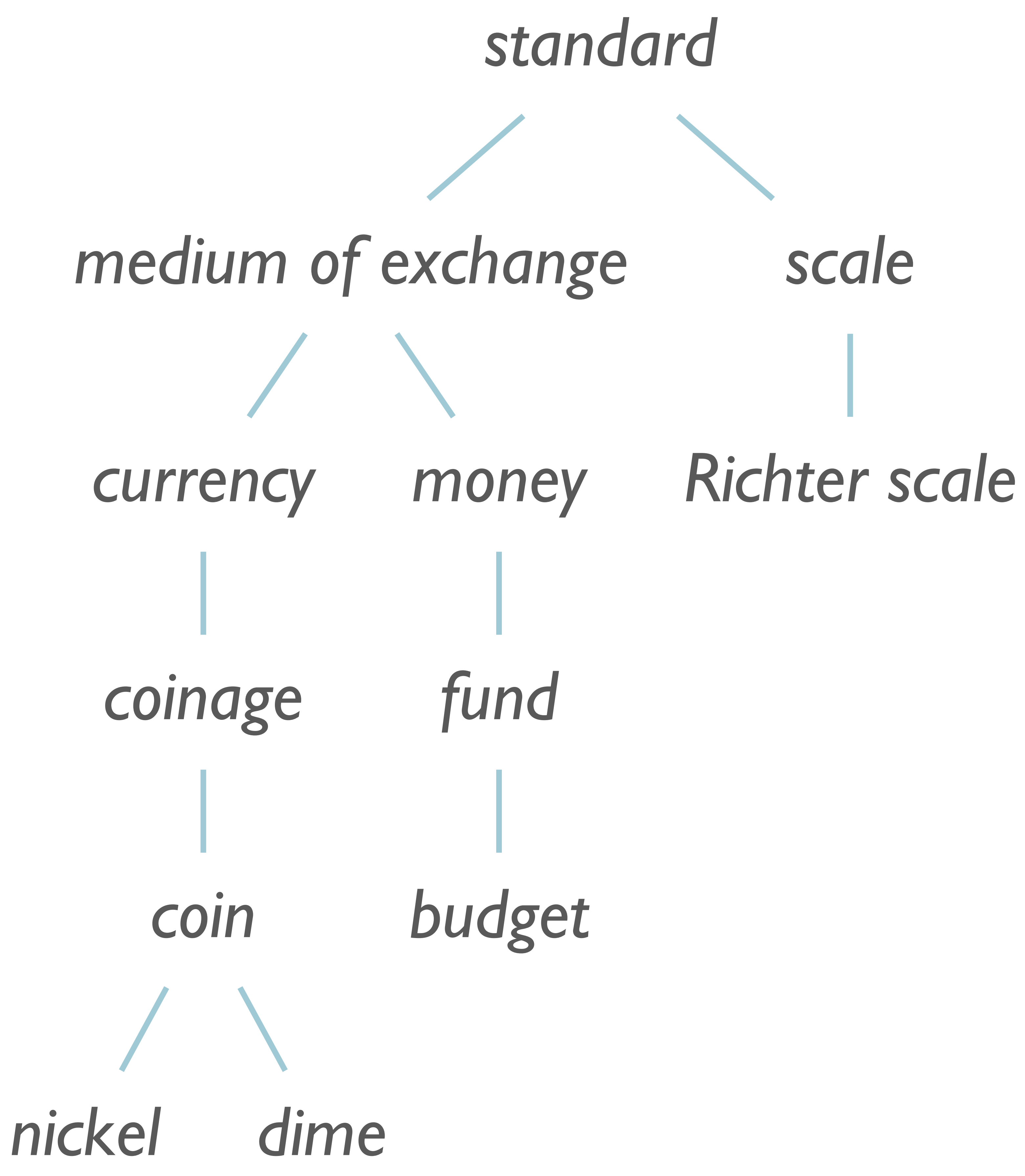
- Information content of node (concept c)
  - $IC(c) = -\log P(c)$
  - As probability of encountering c increases, informativeness decreases
- Least common subsumer (LCS):
  - Lowest node in hierarchy subsuming 2 nodes

Information Content-Based Similarity Measures

- Information content of node (concept c)
  - $IC(c) = -\log P(c)$
  - As probability of encountering c increases, informativeness decreases
- Least common subsumer (LCS):
  - Lowest node in hierarchy subsuming 2 nodes
- Similarity measure
  - $sim_{resnik}(c_1, c_2) = -\log\ P(LCS(c_1, c_2))$
  - The more specific the LCS concept, the more similar $c_1,\ c_2$.

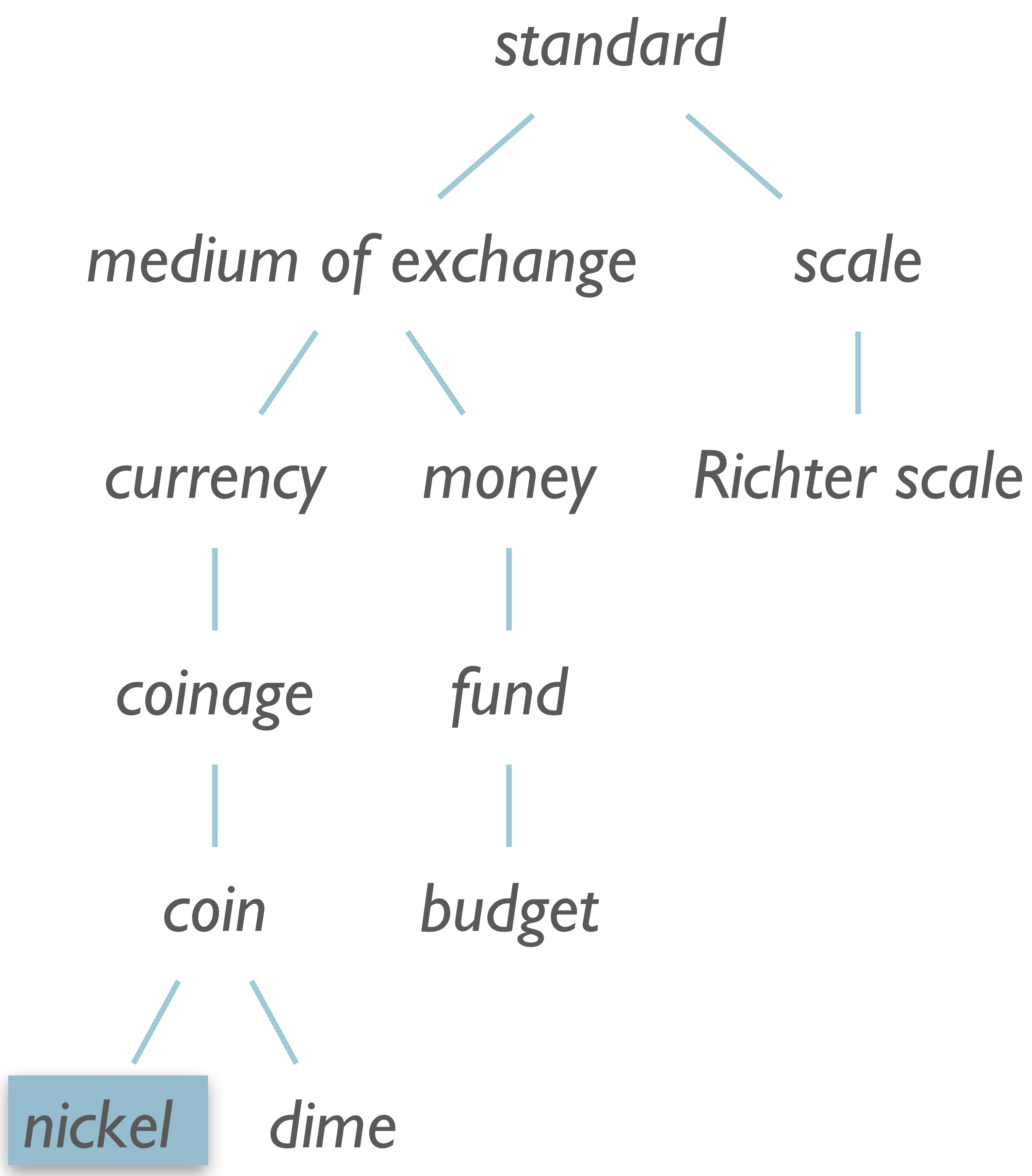
Least Common Subsumer

- LCS(*nickel*, *dime*) = *coin*
- LCS(*nickel*, *budget*) = *medium of exchange*



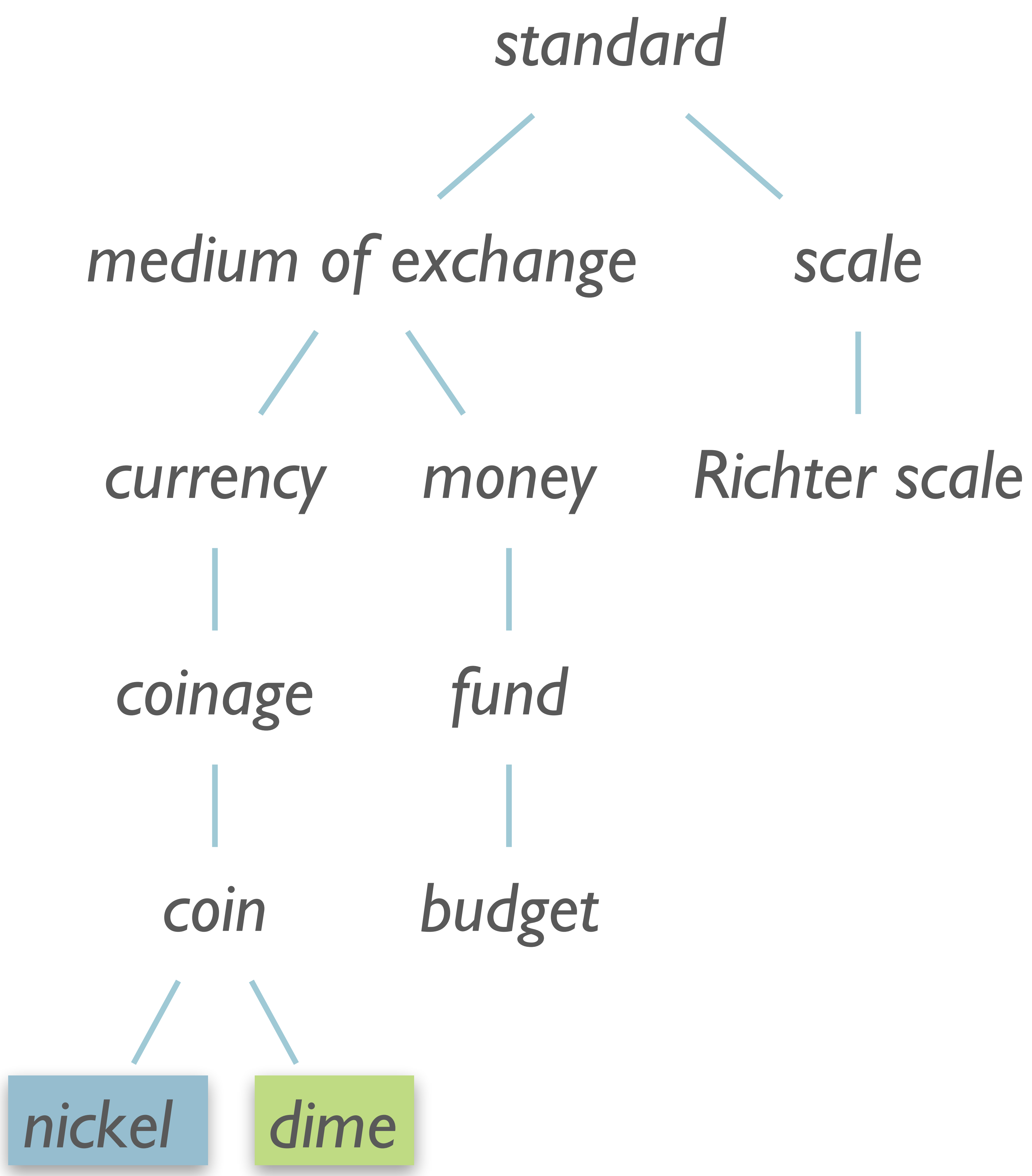
Least Common Subsumer

- LCS(*nickel*, *dime*) = *coin*
- LCS(*nickel*, *budget*) = *medium of exchange*



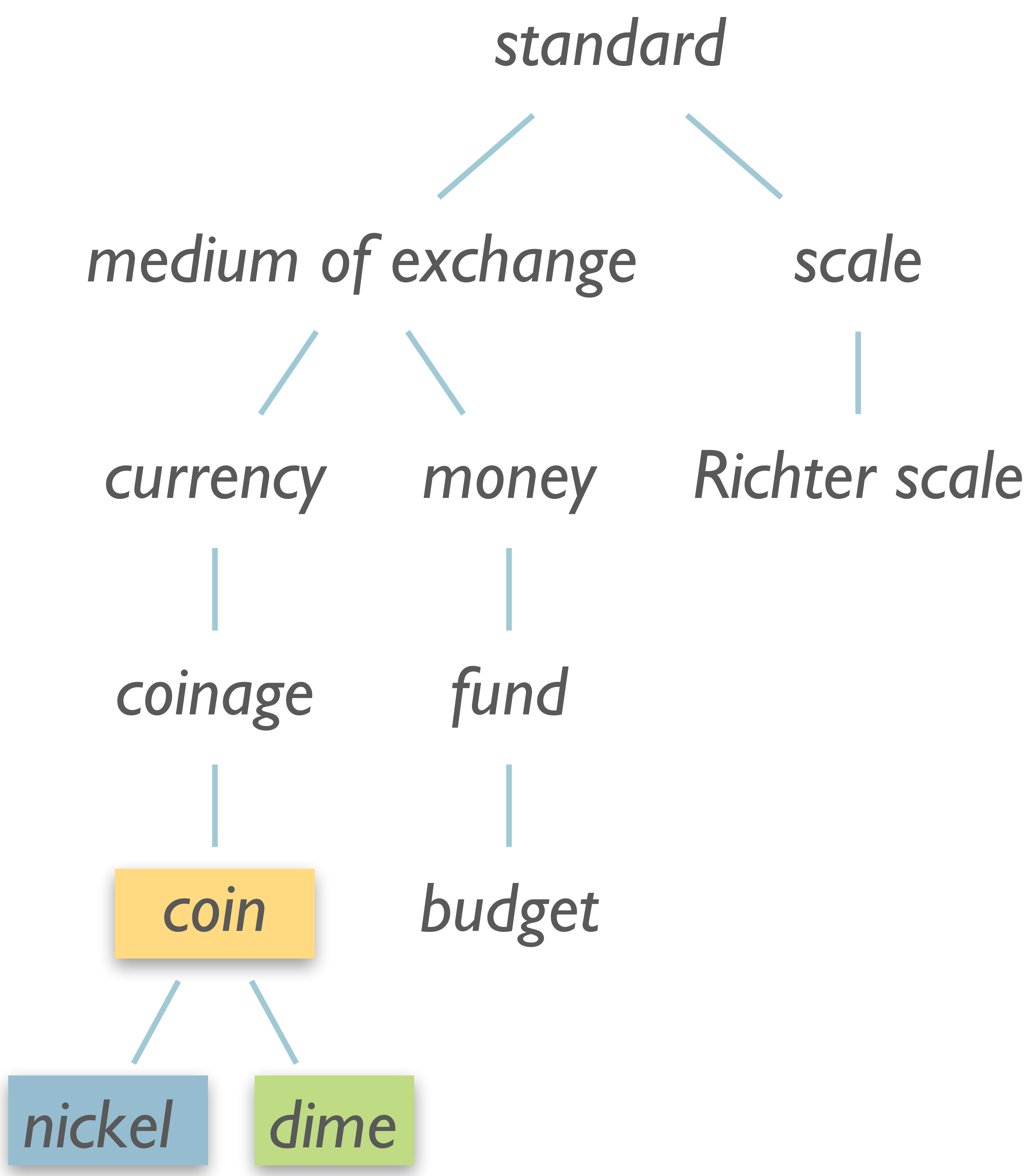
Least Common Subsumer

- LCS(*nickel*, *dime*) = *coin*
- LCS(*nickel*, *budget*) = *medium of exchange*



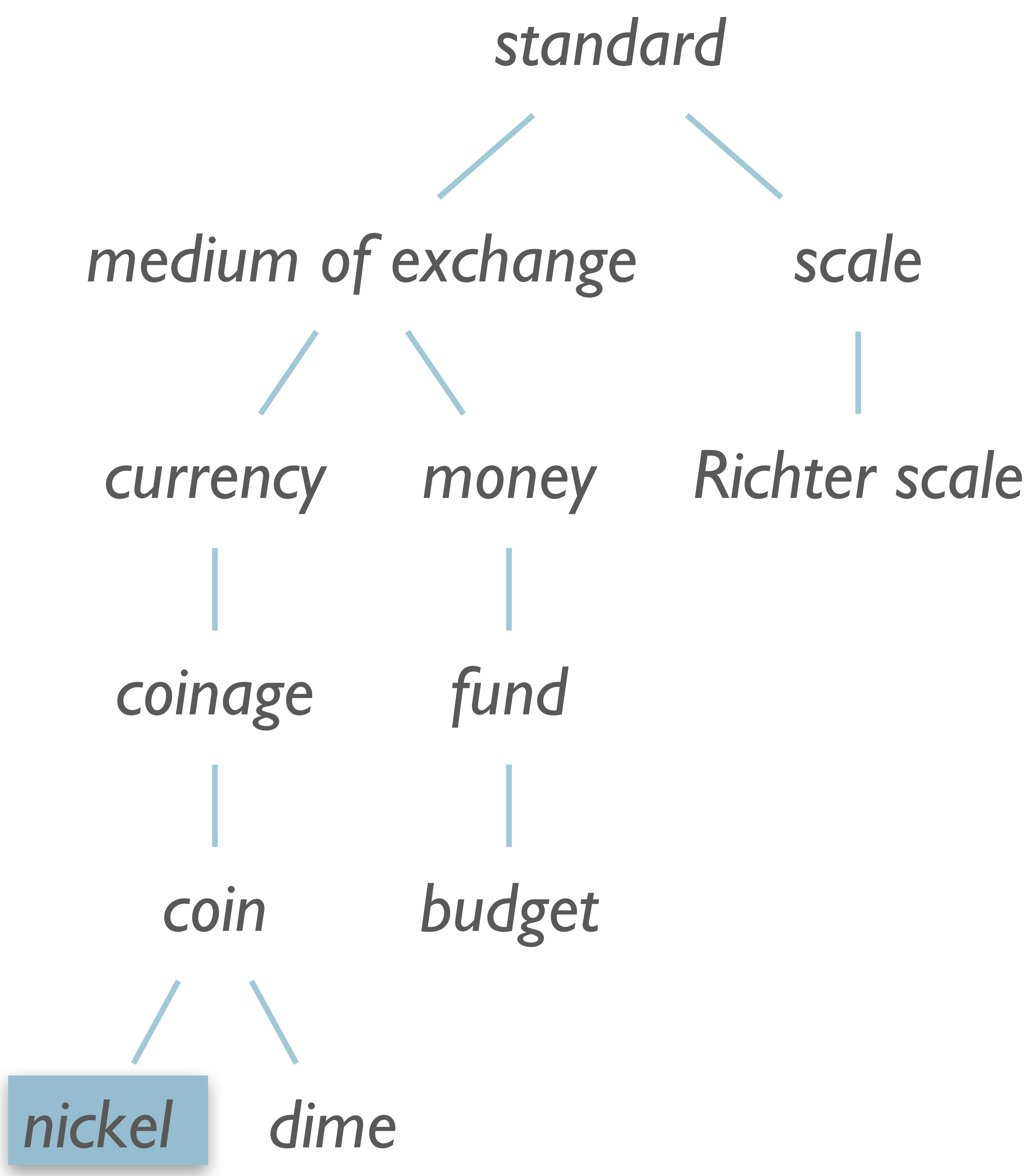
Least Common Subsumer

- LCS(*nickel*, *dime*) = *coin*
- LCS(*nickel*, *budget*) = *medium of exchange*



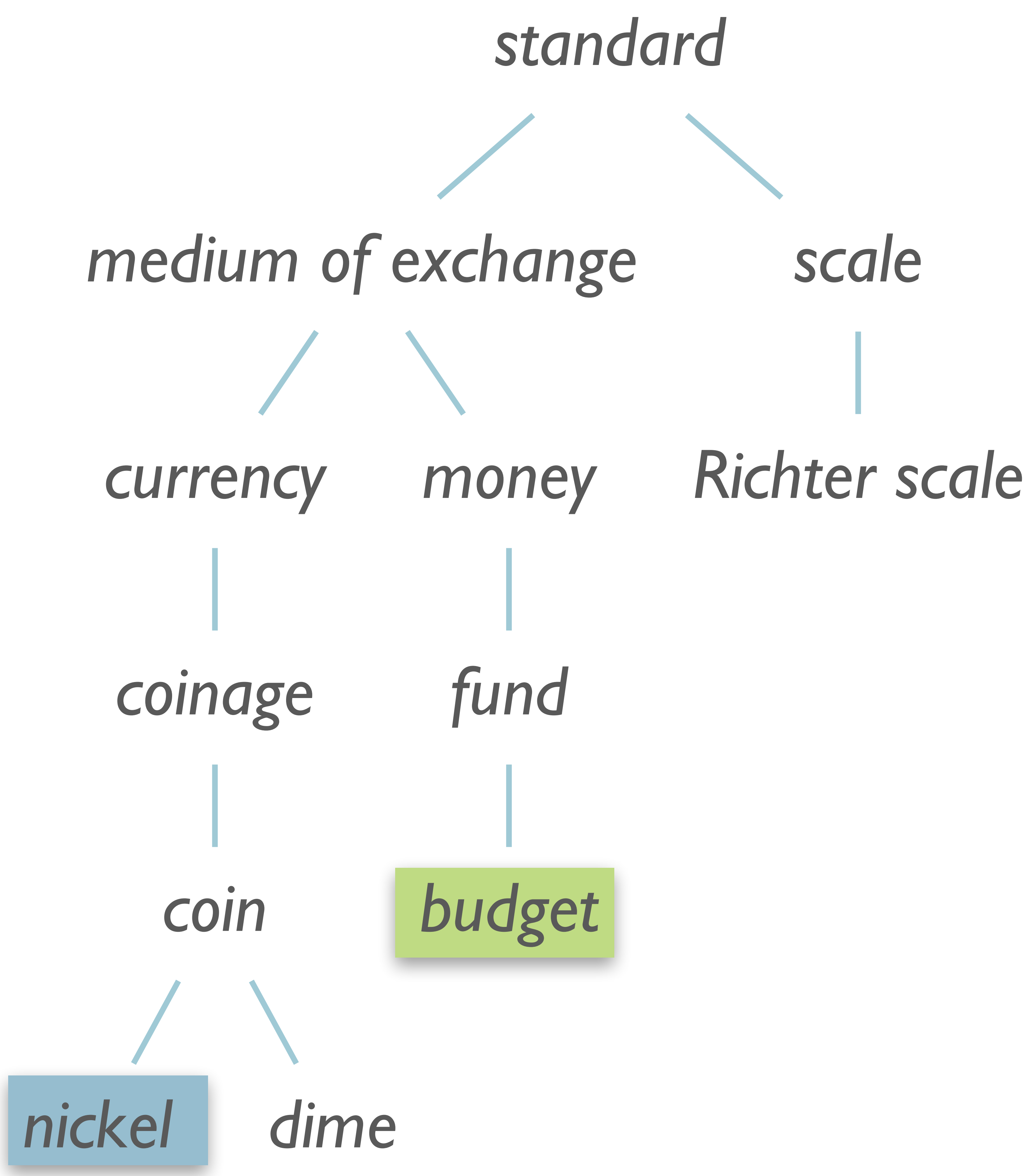
Least Common Subsumer

- LCS(*nickel*, *dime*) = *coin*
- LCS(*nickel*, *budget*) = *medium of exchange*



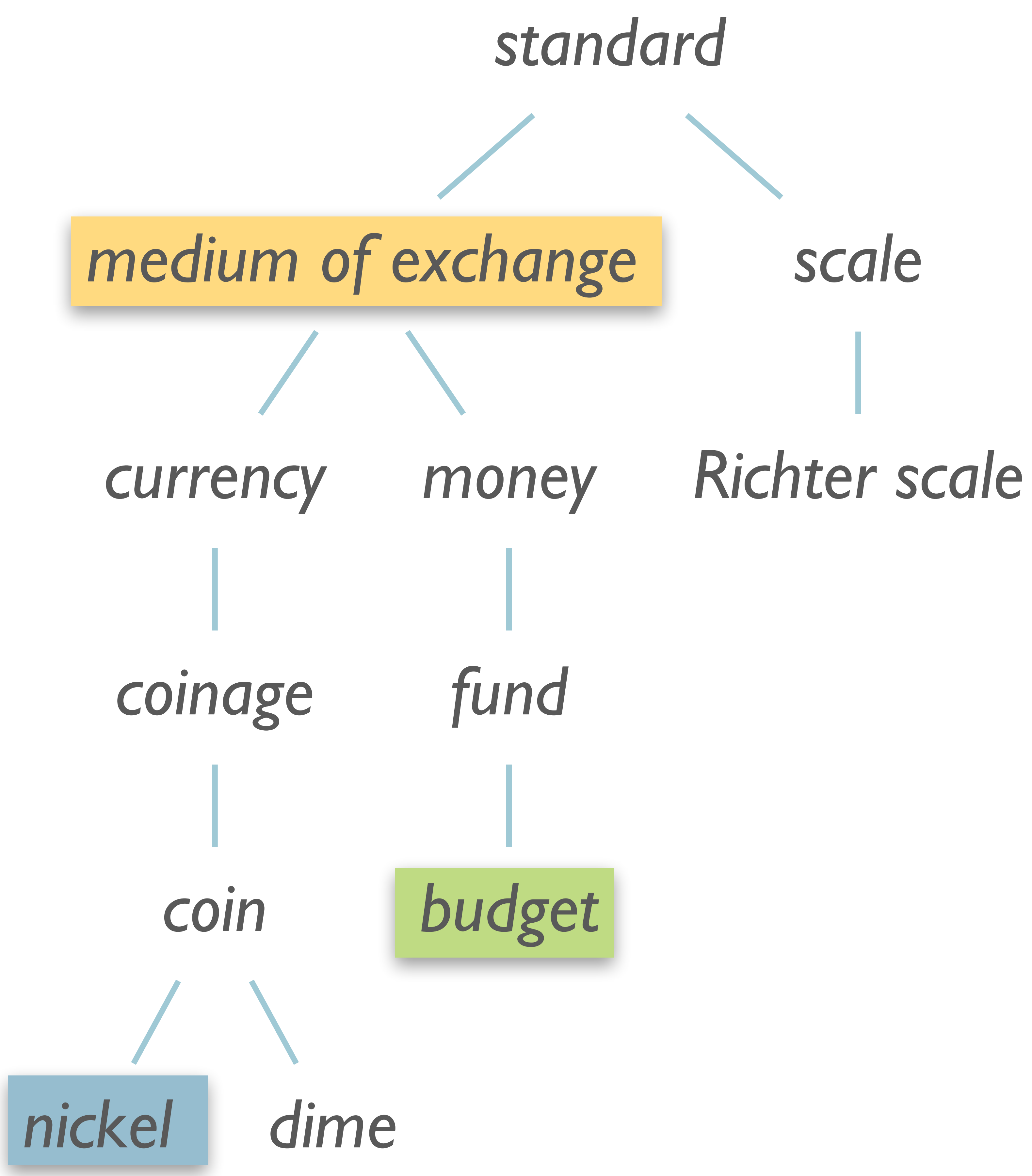
Least Common Subsumer

- LCS(*nickel*, *dime*) = *coin*
- LCS(*nickel*, *budget*) = *medium of exchange*



Least Common Subsumer

- LCS(*nickel*, *dime*) = *coin*
- LCS(*nickel*, *budget*) = *medium of exchange*



The Plant Example Again

- There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.
- The Paulus company was founded in 1938. Since those days the product range has been the subject of constant expansions and is brought up continuously to correspond with the state of the art. We're engineering, manufacturing, and commissioning world-wide ready-to-run **plants** packed with our comprehensive know-how.

Application to WSD

- **Calculate Informativeness**
  - For each node in WordNet:
    - Sum occurrences of concept and all children
    - Compute *Information Content* for each node of WordNet

Application to WSD

- **Disambiguate with WordNet**
  - Assume set of words in context: {*animals*, *rainforest, species*}
  - Find *Most Informative* Least Common Subsumer
    - for target word, context word
  - Increment count for sense subsumed by this concept
  - Select sense with highest vote

Thesaurus Similarity Issues

- Coverage:
  - Few languages have large thesauri
  - Few languages have large sense-tagged corpora
- Thesaurus design:
  - Works well for noun *IS-A* hierarchy
  - Verb hierarchy shallow, bushy, less informative

Resnik Similarity

Algorithm

Given $W=\{w_i,\ldots,w_n\}$, a set of nouns

```
for i and j=1 to n, with i < j
   v_{i,j}=wsim(w_i,w_j)
   c_{i,j}=the most informative subsumer for w_i and w_j

   for k=1 to num_senses(w_i)
      if c_{i,j} is an ancestor of sense_{i,k}
         increment support[i,k] by v_{i,j}

   for k'=1 to num_senses(w_j)
      if c_{i,j} is an ancestor of sense_{j,k'}
         increment_support[j,k'] by v_{i,j}

   increment normalization[i] by v_{i,j}
   increment normalization[j] by v_{i,j}

for i=1 to n
   for k=1 to num_senses(w_i)
      if (normalization[i] > 0.0)
         γ_{i,k}=support[i,k]/normalization[i]
      else
         γ_{i,k}=1/num_senses[w_i]
```

Resnik 1999, sec 5.1
[also on website]

Algorithm

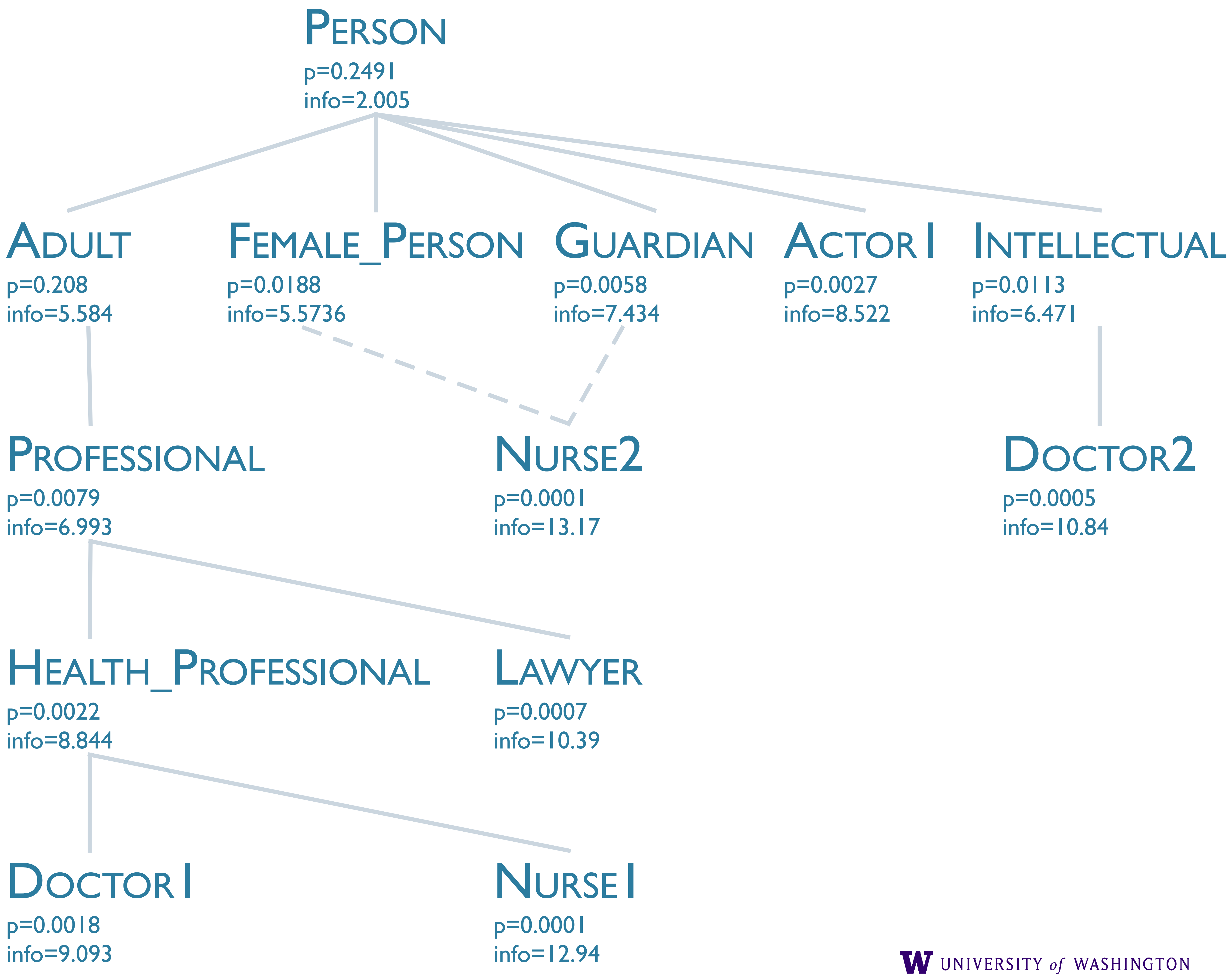
```
Given W={wᵢ,…,wₙ}, a set of nouns
for i=1 to n, and input word w₀
   v₀,ᵢ=wsim(w₀,wᵢ)
   c₀,ᵢ=the most informative subsumer for w₀ and wᵢ

   for k=1 to num_senses(wᵢ)
      of c₀,ᵢ is an ancestor of senseᵢ,ₖ
         increment support[i,k] by v₀,ᵢ

   for k'=1 to num_senses(w₀)
      if c₀,ᵢ is an ancestor of senseₖ'
         increment_support[0,k'] by v₀,ᵢ

   increment normalization[i] by v₀,ᵢ

for i=1 to n
   for k=1 to num_senses(wᵢ)
      if (normalization[i] > 0.0)
         γᵢ,ₖ=support[i,k]/normalization[i]
      else
         γᵢ,ₖ=1/num_senses[wᵢ]
```

Resnik Similarity

- Calculate:

$$sim_{word}(w_1, w_2) = \max_{c_1, c_2}\left(sim_{concept}(c_1, c_2)\right)$$

- Let's try
  - $sim_{word}(doctor,\ nurse)$



Via Resnik (1999) — p. 96

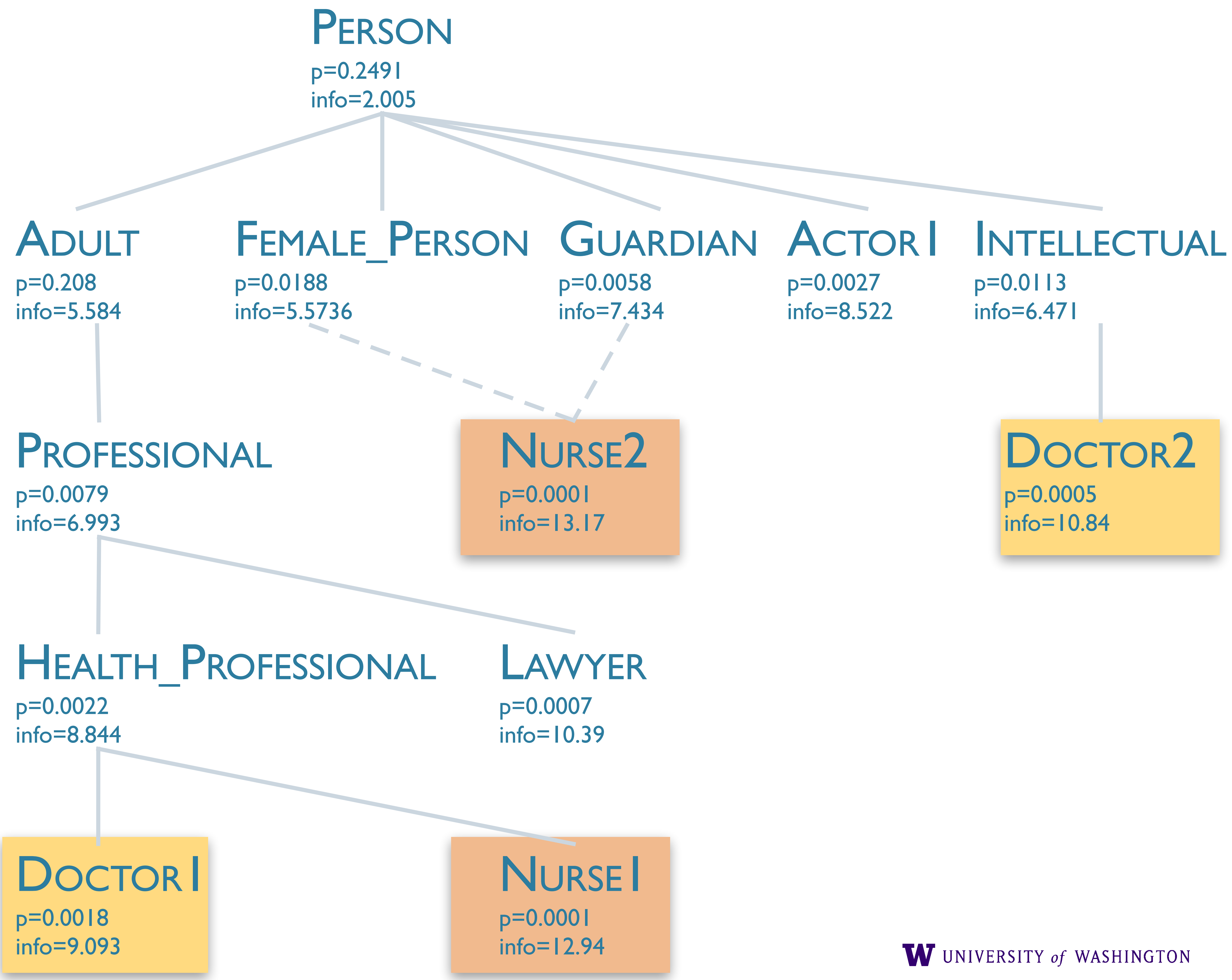
Resnik Similarity

- Calculate:

$$sim_{word}(w_1, w_2) = \max_{c_1, c_2} \left(sim_{concept}(c_1, c_2) \right)$$

- Let's try
  - $sim_{word}(doctor, nurse)$
- $sim_{concept}(c_1, c_2)$
  - Get IC of LCS

c_1	c_2	LCS	$sim(c_1, c_2)$



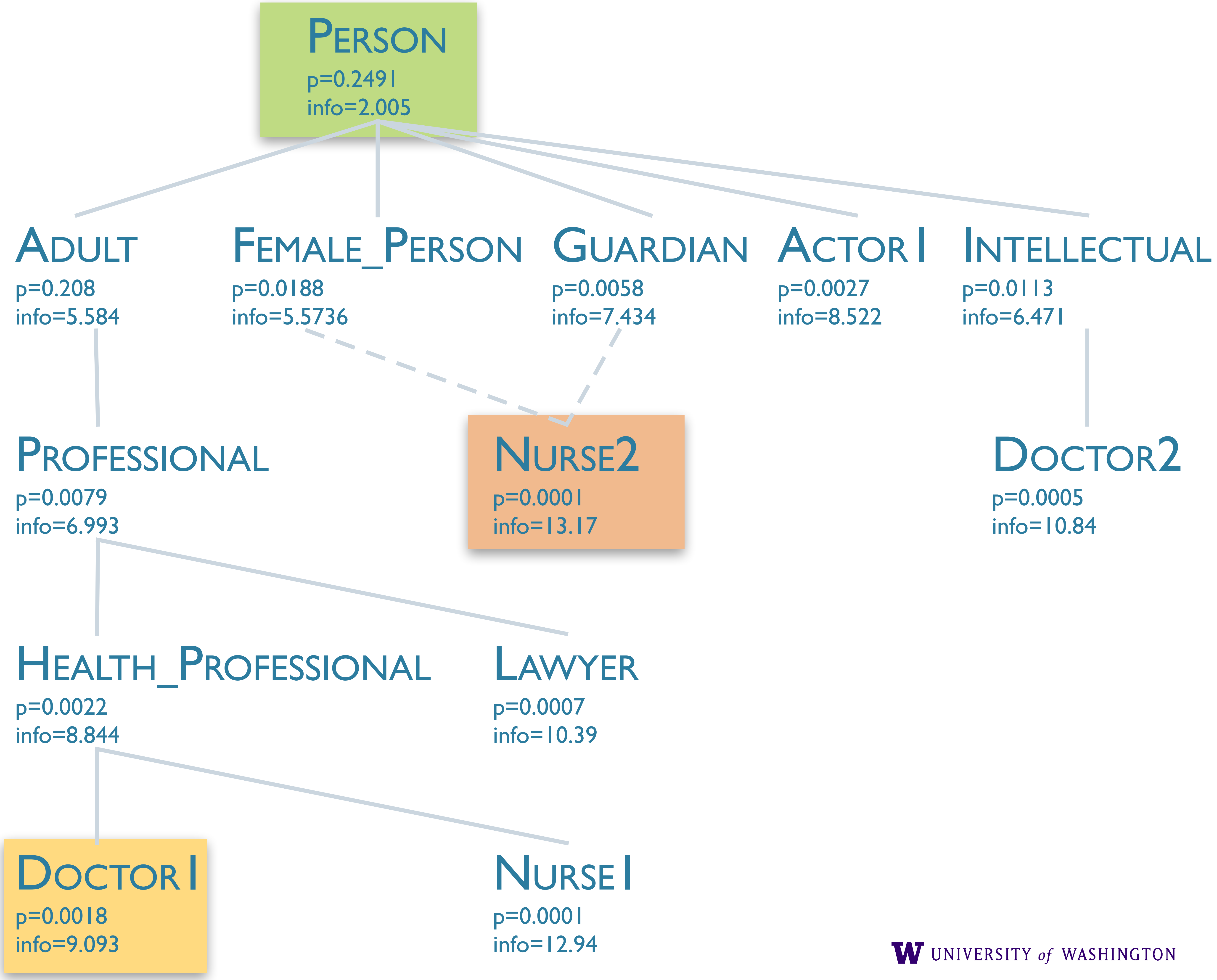
Resnik Similarity

- Calculate:

$$sim_{word}(w_1, w_2) = \max_{c_1, c_2} \left(sim_{concept}(c_1, c_2) \right)$$

- Let's try
  - $sim_{word}(doctor, nurse)$
- $sim_{concept}(c_1, c_2)$
  - Get IC of LCS

c_1	c_2	LCS	$sim(c_1,c_2)$
DOCTOR1	NURSE2	PERSON	2.005



Via Resnik (1999) — p. 96

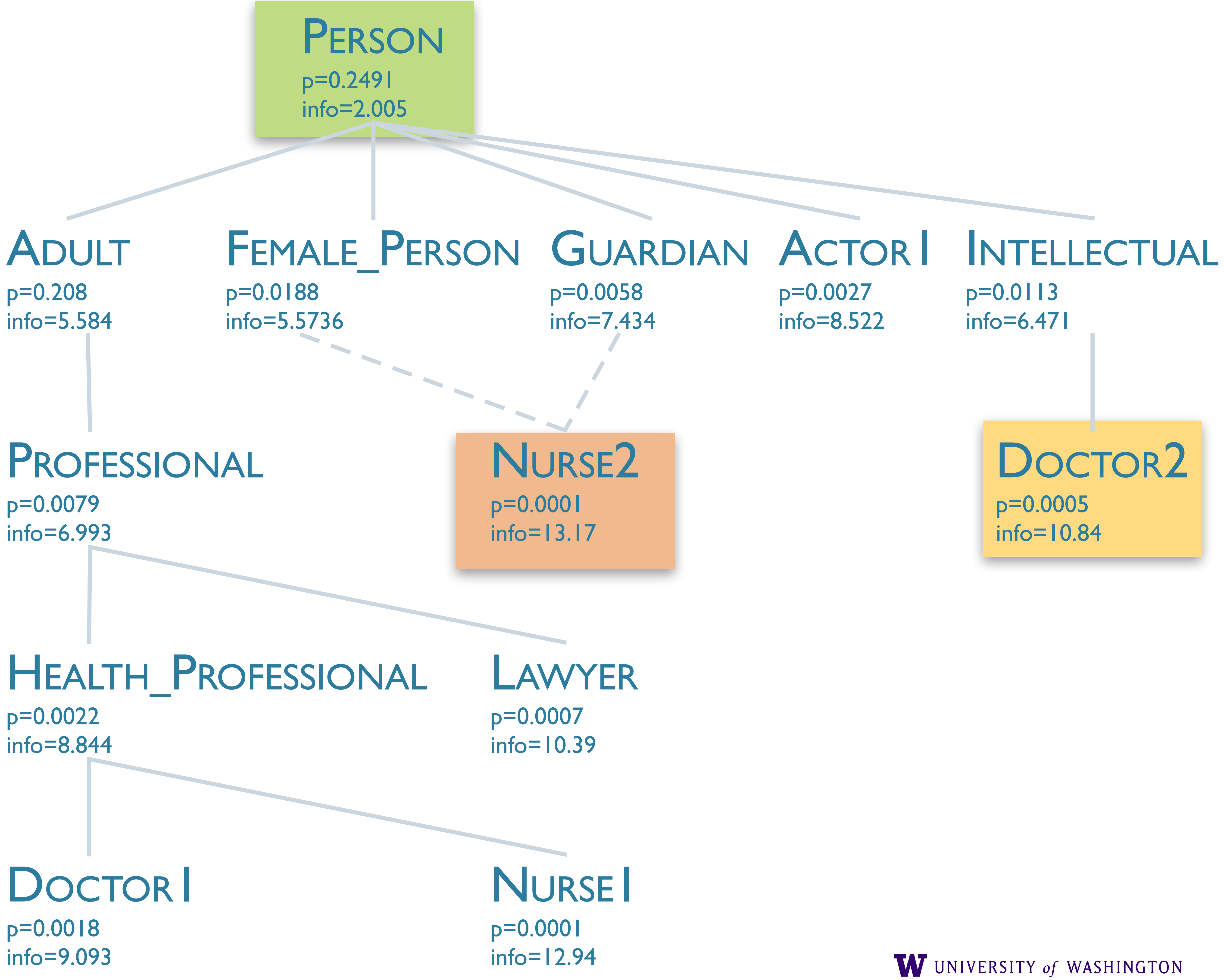
Resnik Similarity

- Calculate:

$$sim_{word}(w_1, w_2) = \max_{c_1, c_2} \left(sim_{concept}(c_1, c_2) \right)$$

- Let's try
  - $sim_{word}(doctor, nurse)$
- $sim_{concept}(c_1, c_2)$
  - Get IC of LCS

c_1	c_2	LCS	$sim(c_1,c_2)$
DOCTOR1	NURSE2	PERSON	2.005
DOCTOR2	NURSE2	PERSON	2.005



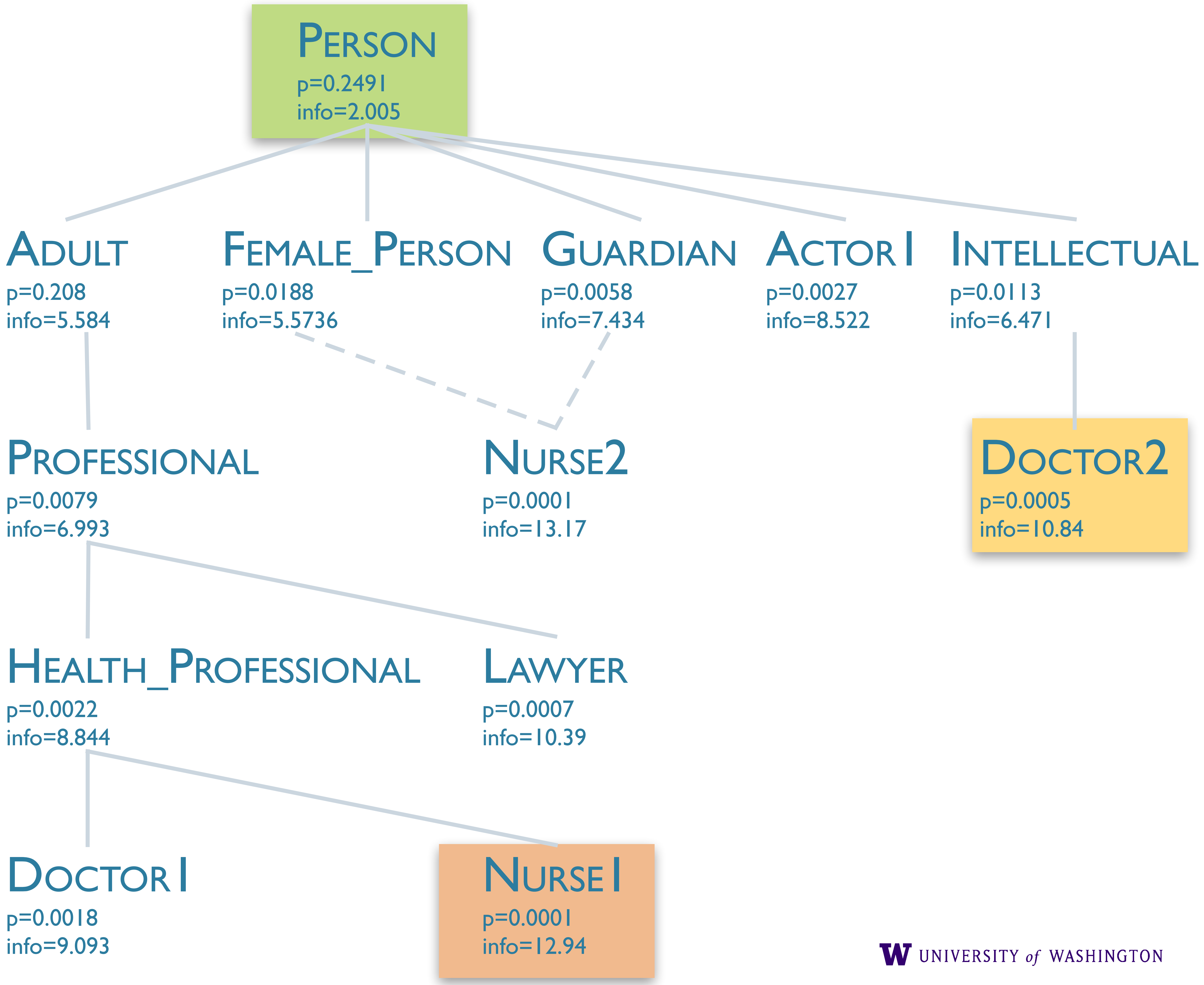
Resnik Similarity

- Calculate:

$$sim_{word}(w_1, w_2) = \max_{c_1, c_2} \left(sim_{concept}(c_1, c_2) \right)$$

- Let's try
  - $sim_{word}(doctor, nurse)$
- $sim_{concept}(c_1, c_2)$
  - Get IC of LCS

c_1	c_2	LCS	$sim(c_1,c_2)$
DOCTOR1	NURSE2	PERSON	2.005
DOCTOR2	NURSE2	PERSON	2.005
DOCTOR2	NURSE1	PERSON	2.005



Via Resnik (1999) — p. 96

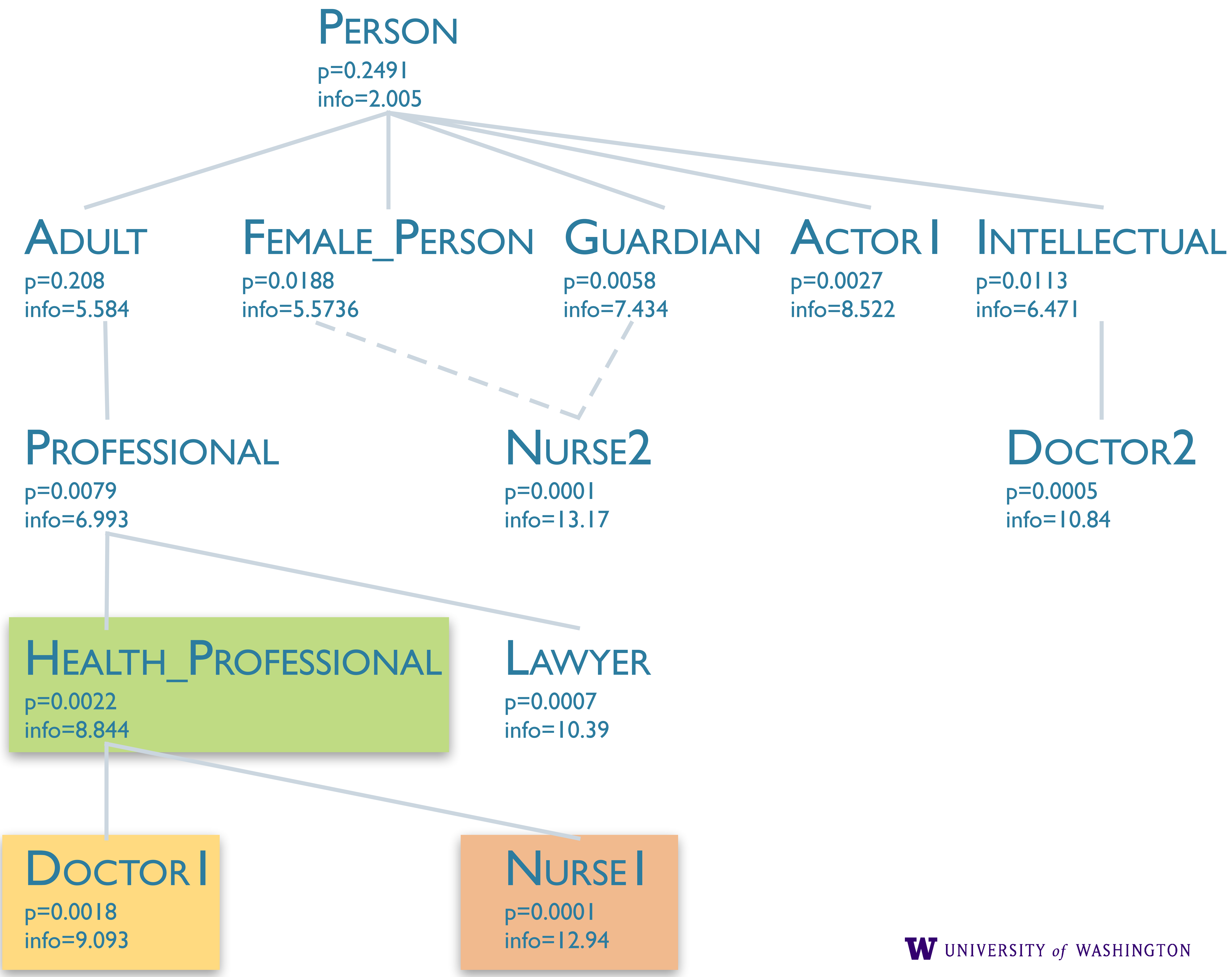
Resnik Similarity

- Calculate:

$$sim_{word}(w_1, w_2) = \max_{c_1,c_2} \left(sim_{concept}(c_1, c_2) \right)$$

- Let's try
  - $sim_{word}(doctor, nurse)$
- $sim_{concept}(c_1, c_2)$
  - Get IC of LCS

c_1	c_2	LCS	sim(c_1,c_2)
Doctor1	Nurse2	Person	2.005
Doctor2	Nurse2	Person	2.005
Doctor2	Nurse1	Person	2.005
Doctor1	Nurse1	Health_Professional	8.844



Via Resnik (1999) — p. 96

Resnik WSD: Choosing a Sense

- doctor — nurse, lawyer, accountant, scholar, minister
- We'll get:
  - {DOCTOR1, NURSE1} ⊂ HEALTH_PROFESSIONAL = 8.844
  - {DOCTOR1, LAWYER1} ⊂ PROFESSIONAL + 6.993 = 15.837
  - {DOCTOR1, ACCOUNTANT1} ⊂ PROFESSIONAL + 6.993 = **22.83**
  - {DOCTOR2, SCHOLAR1} ⊂ INTELLECTUAL = 6.471
  - {DOCTOR2, MINISTER1} ⊂ INTELLECTUAL + 6.471 = **12.942**
- DOCTOR1 with 22.83 of "support"
- DOCTOR2 with 12.942 of "support"
  - Select DOCTOR1 by majority vote.

Compositional and Lexical Semantics

The Meaning of "Life"

Foreword

In the spring of 1976, Terry Parsons and Barbara Partee taught a course on Montague grammar, which I attended. On the second to the final day of class, Terry went around the room asking the students if there were any questions at all that remained unanswered, and promised to answer them on the last day of class. I asked if he really meant ANY question at all, which he emphatically said that he meant. As I had encountered a few questions in my lifetime that remained at least partially unresolved, I decided to ask one of them. What is life? What is the meaning of life? After all, Barbara and Terry had promised to provide answers to any question at all.

On the final day of class Barbara wore her Montague grammar T-shirt, and she and Terry busied themselves answering our questions. At long last, they came to my question. I anticipated a protracted and involved answer, but their reply was crisp and succinct. First Barbara, chalk in hand, showed me the meaning of life.

^life'

Terry then stepped up and showed me what life really is.

˅^life'

As we were asked to show on a homework assignment earlier in the year, this is equivalent to: life'.

Leaving me astounded that I had been living in such darkness for all these years, the class then turned to the much stickier problem of pronouns.

Carlson 1977

The Meaning of "Life"

Foreword

In the spring of 1976, Terry Parsons and Barbara Partee taught a course on Montague grammar, which I attended. On the second to the final day of class, Terry went around the room asking the students if there were any questions at all that remained unanswered, and promised to answer them on the last day of class. I asked if he really meant ANY question at all, which he emphatically said that he meant. As I had encountered a few questions in my lifetime that remained at least partially unresolved, I decided to ask one of them. What is life? What is the meaning of life? After all, Barbara and Terry had promised to provide answers to any question at all.

On the final day of class Barbara wore her Montague grammar T-shirt, and she and Terry busied themselves answering our questions. At long last, they came to my question. I anticipated a protracted and involved answer, but their reply was crisp and succinct. First Barbara, chalk in hand, showed me the meaning of life.

^life' ← `\w.\x.life(w,x)`

Terry then stepped up and showed me what life really is.

ˇ^life'

As we were asked to show on a homework assignment earlier in the year, this is equivalent to: life'.

Leaving me astounded that I had been living in such darkness for all these years, the class then turned to the much stickier problem of pronouns.

Carlson 1977

Two “Approaches” to Meaning

- Compositional / logical semantics:
  - Verb → ‘booked’ {λW.λz.W(∃eBooked(e) ∧ Booker(e,z) ∧ BookedThing(e,y))}
- Lexical semantics:
  - booked: [0.1234, 0.4, 0.269, …]
- Generating good *sentence representations*, either by integrating these two approaches or enriching the distributional approach, is a major area of current work in computational semantics.

HW #8

Implementation

- Implement a simplified version of Resnik's "Associating Word Senses with Noun Groupings"
- Select a sense for the probe word, given group
  - Rather than all words as in the algorithm in the paper
- For each pair (probe, $noun_i$)
  - Loop over sense pairs to find MIS (Most informative sense), similarity value *v*
  - Update each sense of probe descended from MIS, with *v*
- Select highest scoring sense of probe
- Repeat noun-pair correlation with Resnik similarity

Components

- Similarity measure:
  - IC:
    - `/corpora/nltk/nltk-data/corpora/wordnet_ic/ic-brown-resnik-add1.dat`
  - NLTK accessor:
    - `wnic = nltk.corpus.wordnet_ic.ic('ic-brown-resnik-add1.dat')`
  - Note: Uses WordNet 3.0

Components

```
>>> from nltk.corpus import *
>>> brown_ic = wordnet_ic.ic('ic-brown-resnik-add1.dat')
>>> wordnet.synsets('artifact')
[Synset('artifact.n.01')]

>>> wordnet.synsets('artifact')[0].name
 'artifact.n.01'

>>> artifact = wordnet.synset('artifact.n.01')
from nltk.corpus.reader.wordnet import information_content

>>> information_content(artifact, brown_ic)
2.4369607933293391
```

Components

- Hypernyms:

```
>>>wn.synsets('artifact')[0].hypernyms()
[Synset('whole.n.02')]
```

- Common hypernyms:

```
>>> hat = wn.synsets('hat')[0]
>>> glove = wn.synsets('glove')[0]
>>> hat.common_hypernyms(glove)
[Synset('object.n.01'), Synset('artifact.n.01'),
Synset('whole.n.02'), Synset('physical_entity.n.01'),
Synset('entity.n.01')]
```

Components

- WordNet API
  - NLTK: **Strongly** suggested
  - Others exist, but no “warranty”!
- http://www.nltk.org/howto/wordnet.html
- http://www.nltk.org/api/nltk.corpus.reader.html#module-nltk.corpus.reader.wordnet

Note

- You can use supporting functionality, e.g.
  - `common_hypernyms`, `full_hypernyms`, etc
- You can NOT just use the built-in
  - `resnik_similarity`
  - `least_common_hypernym`, etc
- If unsure about acceptability, just ask!