CS 6120/CS 4120: Natural Language Processing

Instructor: Prof. Lu Wang

College of Computer and Information Science

Northeastern University

Webpage: www.ccs.neu.edu/home/luwang

Logistics

- Assignment 1 is released, due on Feb 13th, 11:59pm

- Quiz solution is posted on Blackboard.
  - For quizzes of 1,3,5,7, please see Nikhil if you have questions wrt grading.
  - For quizzes of 0,2,4,6,8, please see Parmeet.

- All questions related to the course should be posted on Piazza
  - http://piazza.com/northeastern/spring2019/cs6120/home

- Tentative final exam time: 1-3pm, April 26. Location: TBD

Outline

- Word Senses and Word Relations
- Word Similarity
- Word Sense Disambiguation

Terminology: lemma and wordform

- A **lemma** or **citation form**
  - Same stem, part of speech, rough semantics
- A **wordform**
  - The inflected word as it appears in text

Wordform	Lemma
banks	bank
sung	sing
duermes	dormir

Lemmas have senses

- One lemma “bank” can have many meanings:

Sense 1: • `…a` **bank**$_1$ `can hold the investments in a custodial account…`

Sense 2: • `“…as agriculture burgeons on the east` **bank**$_2$ `the river will shrink even more”`

- **Sense** (or **word sense**)
  - A discrete representation of an aspect of a word’s meaning.
- The lemma **bank** here has two senses

Homonymy

Homonyms: words that share a form (spell or sound alike) but have unrelated, distinct meanings:

- bank_1: financial institution, bank_2: sloping land
- bat_1: club for hitting a ball, bat_2: nocturnal flying mammal

1. Homographs (bank/bank, bat/bat)
2. Homophones:
    1. Write and right
    2. Piece and peace

Homonymy causes problems for NLP applications

- Information retrieval
  - “bat care”
- Machine Translation
  - bat: murciélago (animal) or bate (for baseball)
- Text-to-Speech
  - bass (stringed instrument) vs. bass (fish)

- 1. The **bank** was constructed in 1875 out of local red brick.
- 2. I withdrew the money from the **bank**

Polysemy

- 1. The **bank** was constructed in 1875 out of local red brick.
- 2. I withdrew the money from the **bank**
- Are those the same sense?
  - Sense 1: "The building belonging to a financial institution"
  - Sense 2: "A financial institution"
- A **polysemous** word has **related** meanings
  - Most non-rare words have multiple meanings

Metonymy or Systematic Polysemy: A systematic relationship between senses

- Lots of types of polysemy are systematic
  - `School, university, hospital`
  - All can mean the institution or the building.
- A systematic relationship:
  - Building ⟺ Organization
- Other such kinds of systematic polysemy:

Author `(Jane Austen wrote Emma)`

⟺ Works of Author `(I love Jane Austen)`

Tree `(Plums have beautiful blossoms)`

⟺ Fruit `(I ate a preserved plum)`

How do we know when a word has more than one sense?

- The "zeugma" test: Two senses of `serve`?
  - `Which flights` **`serve`** `breakfast?`
  - `Does Lufthansa` **`serve`** `Philadelphia?`
  - Does Lufthansa serve breakfast and San Jose?
- Since this conjunction sounds weird,
  - we say that these are **two different senses of "serve"**

Synonyms

- Words that have the same meaning in some or all contexts.
  - filbert / hazelnut
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - Water / H_20
- Two words are synonyms if they can be substituted for each other in all situations (strict/perfect definition).

Synonyms

- But there are few (or no) examples of perfect synonymy.
  - Even if many aspects of meaning are identical
  - Still may not preserve the acceptability based on notions of politeness, slang, register, genre, etc.
- Example:
  - Water/H_20
  - Big/large
  - Brave/courageous

Synonymy is a relation between senses rather than words

- Consider the words *big* and *large*
- Are they synonyms?
  - How **big** is that plane?
  - Would I be flying on a **large** or small plane?
- How about here:
  - Miss Nelson became a kind of **big** sister to Benjamin.
  - Miss Nelson became a kind of **large** sister to Benjamin.
- Why?
  - *big* has a sense that means being older, or grown up
  - *large* lacks this sense

Antonyms

- Senses that are opposites with respect to one feature of meaning
- Otherwise, they are very similar!

```
dark/light   short/long    fast/slow rise/fall
hot/cold        up/down            in/out
```

- More formally: antonyms can
  - define a binary opposition or be at opposite ends of a scale
    - `long/short, fast/slow`
  - Be **reversives**:
    - `rise/fall, up/down`

Hyponymy and Hypernymy

- One sense is a **hyponym** of another if the first sense is more specific, denoting a subclass of the other
  - *car* is a hyponym of *vehicle*
  - *mango* is a hyponym of *fruit*
- Conversely **hypernym/superordinate** ("hyper is super")
  - *vehicle* is a **hypernym** of *car*
  - *fruit* is a hypernym of *mango*

Superordinate/hypernym	vehicle	fruit	furniture
Subordinate/hyponym	car	mango	chair

Hyponymy more formally

- Extensional:
  - The class denoted by the superordinate extensionally includes the class denoted by the hyponym
- Entailment:
  - A sense A is a hyponym of sense B if *being an A* entails *being a B*
- Hyponymy is usually transitive
  - (A hypo B and B hypo C entails A hypo C)
- Another name: the **IS-A hierarchy**
  - A IS-A B (or A ISA B)
  - B **subsumes** A

Superordinate/hypernym	vehicle	fruit	furniture
Subordinate/hyponym	car	mango	chair

Hyponymy more formally

- Extensional:
  - The class denoted by the superordinate extensionally includes the class denoted by the hyponym
- Entailment:
  - A sense A is a hyponym of sense B if *being an A* entails *being a B*
- Hyponymy is usually transitive
  - (A hypo B and B hypo C entails A hypo C)
- Another name: the **IS-A hierarchy**
  - A IS-A B (or A ISA B)
  - B **subsumes** A

Applications in textual entailment or reasoning or machine comprehension

Superordinate/hypernym	vehicle	fruit	furniture
Subordinate/hyponym	car	mango	chair

Hyponyms and Instances

- WordNet (introduced later) has both **classes** and **instances**.
- An **instance** is an individual, a proper noun that is a unique entity
  - `San Francisco` is an **instance** of `city`
- But `city` is a class
  - `city` is a **hyponym** of `municipality...location...`

Meronymy

- The part-whole relation
    - A *leg* is part of a *chair*; a *wheel* is part of a *car*.
- *Wheel* is a **meronym** of *car*, and *car* is a **holonym** of *wheel*.

WordNet 3.0

- A hierarchically organized lexical database
- On-line thesaurus + aspects of a dictionary

Category	Unique Strings
Noun	117,798
Verb	11,529
Adjective	22,479
Adverb	4,481

EuroWordNet

- WordNets for
  - Dutch
  - Italian
  - Spanish
  - German
  - French
  - Czech
  - Estonian

Senses of “bass” in Wordnet

Noun

- S: (n) **bass** (the lowest part of the musical range)
- S: (n) **bass**, bass part (the lowest part in polyphonic music)
- **S: (n) bass, basso (an adult male singer with the lowest voice)**
- S: (n) sea bass, **bass** (the lean flesh of a saltwater fish of the family Serranidae)
- S: (n) freshwater bass, **bass** (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
- S: (n) **bass**, bass voice, basso (the lowest adult male singing voice)
- S: (n) **bass** (the member with the lowest range of a family of musical instruments)
- S: (n) **bass** (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

Adjective

- S: (adj) **bass**, deep (having or denoting a low vocal or instrumental range) *"a deep voice"; "a bass voice is lower than a baritone voice"; "a bass clarinet"*

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:

  chump^{1}, fool^{2}, gull^{1}, mark^{9}, patsy^{1}, fall guy^{1}, sucker^{1}, soft touch^{1}, mug^{2}

- Each of **these** senses have this same gloss
  - (Not **every** sense; sense 2 of gull is the aquatic bird)

WordNet Hypernym Hierarchy for "bass"

- S: (n) **bass**, basso (an adult male singer with the lowest voice)
    - *direct hypernym* | ***inherited hypernym*** | *sister term*
        - S: (n) singer, vocalist, vocalizer, vocaliser (a person who sings)
            - S: (n) musician, instrumentalist, player (someone who plays a musical instrument (as a profession))
                - S: (n) performer, performing artist (an entertainer who performs a dramatic or musical work for an audience)
                    - S: (n) entertainer (a person who tries to please or amuse)
                        - S: (n) person, individual, someone, somebody, mortal, soul (a human being) *"there was too much for one person to do"*
                            - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
                                - S: (n) living thing, animate thing (a living (or once living) entity)
                                    - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
                                        - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
                                            - S: (n) physical entity (an entity that has physical existence)
                                                - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

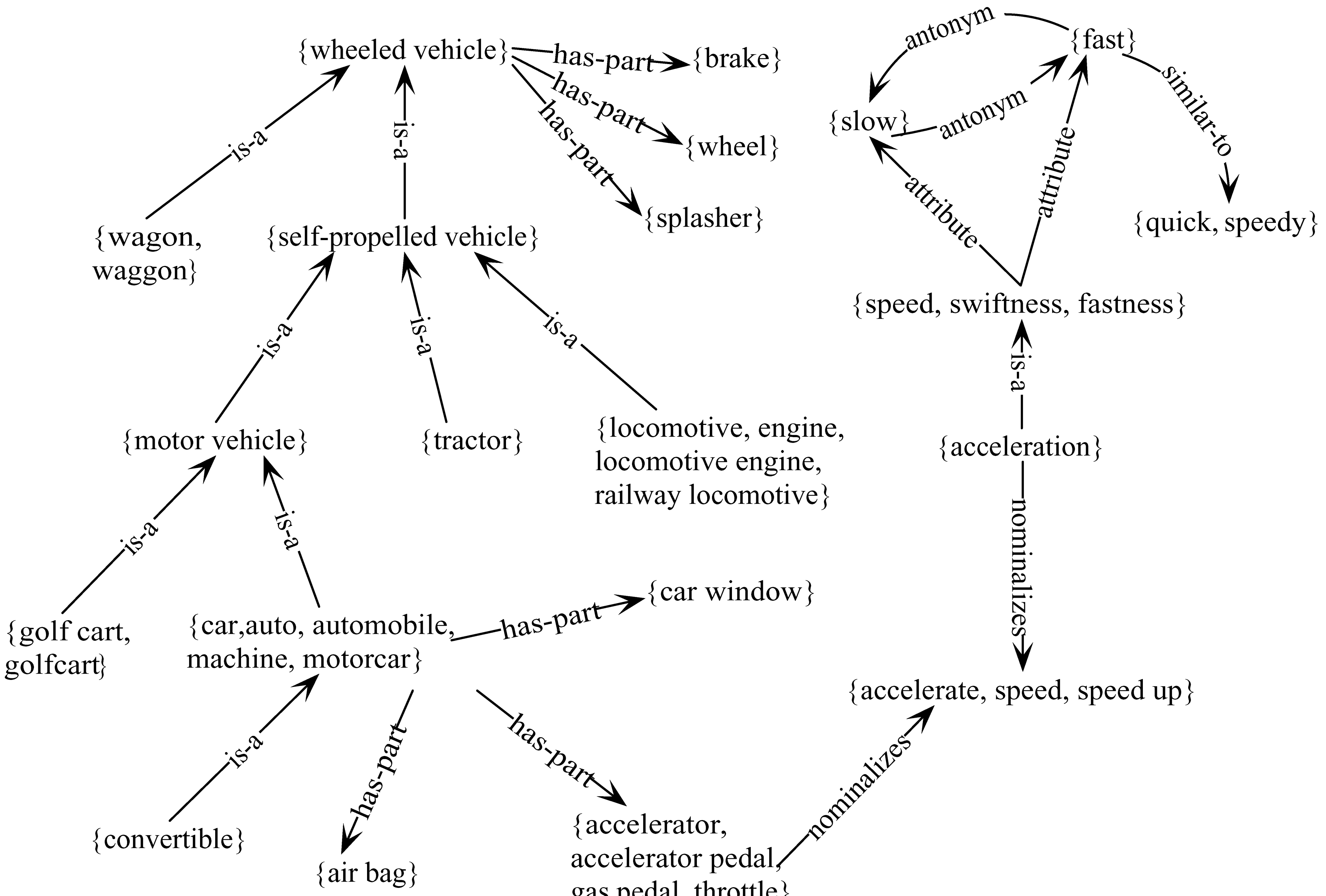
WordNet Noun Relations

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

WordNet Verb Relations

Relation	Definition	Example
Hypernym	From events to superordinate events	*fly*^{9} $\rightarrow$ *travel*^{5}
Troponym	From events to subordinate event (often via specific manner)	*walk*^{1} $\rightarrow$ *stroll*^{1}
Entails	From verbs (events) to the verbs (events) they entail	*snore*^{1} $\rightarrow$ *sleep*^{1}
Antonym	Semantic opposition between lemmas	*increase*^{1} $\Longleftrightarrow$ *decrease*^{1}
Derivationally Related Form	Lemmas with same morphological root	*destroy*^{1} $\Longleftrightarrow$ *destruction*^{1}

WordNet: Viewed as a graph



WordNet 3.0

- Where it is:
  - http://wordnetweb.princeton.edu/perl/webwn
- Libraries
  - Python: WordNet from NLTK
    - http://www.nltk.org/Home
  - Java:
    - JWNL, extJWNL on sourceforge

Outline

- Word Senses and Word Relations
- Word Similarity
- Word Sense Disambiguation

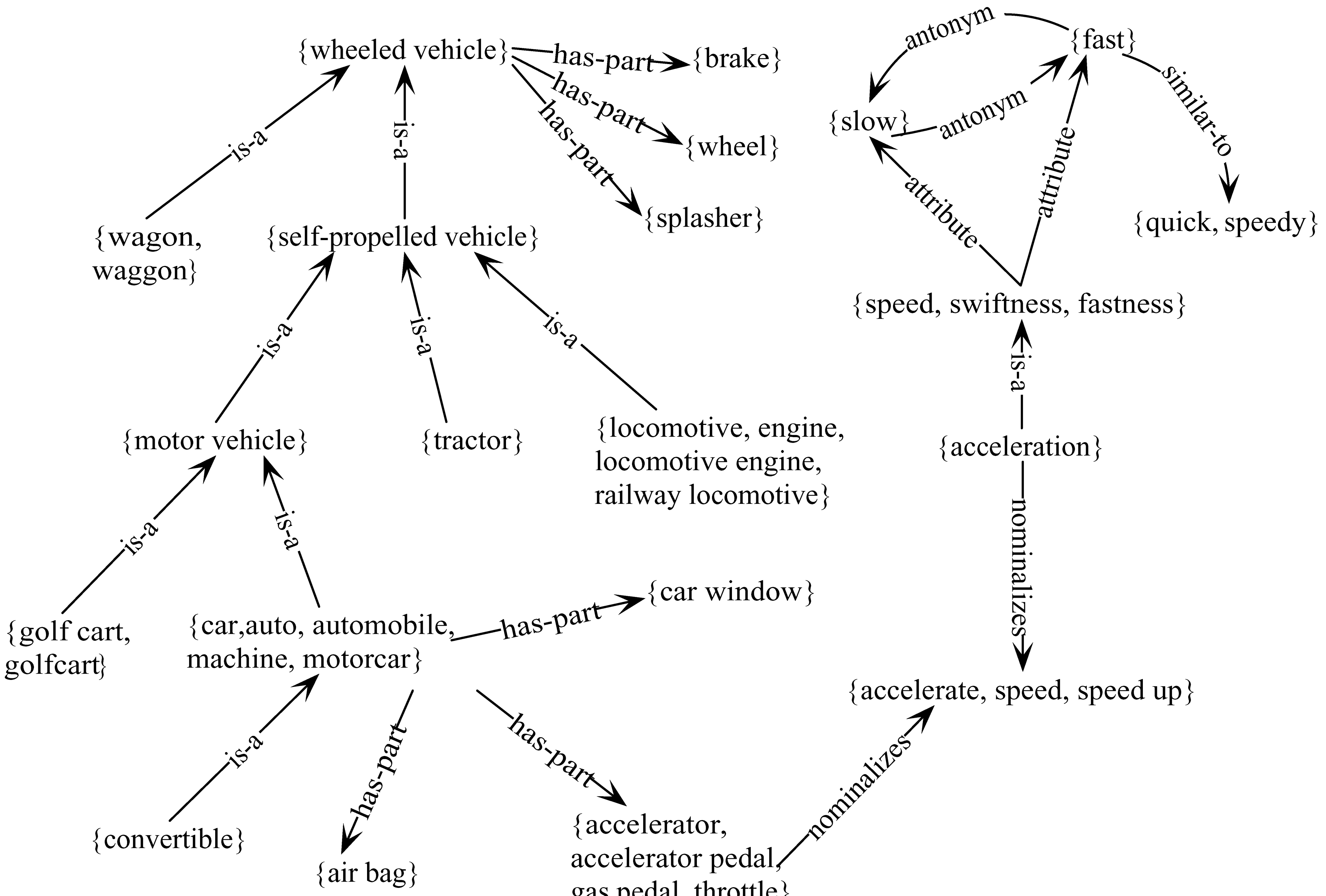
Why word similarity

- A practical component in lots of NLP tasks
  - Question answering
  - Natural language generation
  - Automatic essay grading
  - Plagiarism detection
- A theoretical component in many linguistic and cognitive tasks
  - Historical semantics
  - Models of human word learning
  - Morphology and grammar induction

Word Similarity

- **Synonymy**: a binary relation
  - Two words are either synonymous or not
- **Similarity** (or **distance**): a looser metric (more useful in practice!)
  - Two words are more similar if they share more features of meaning
- Similarity is properly a relation between **senses**
  - Bank^1 is similar to fund^3
  - Bank^2 is similar to slope^5
- But we'll compute similarity over both words and senses

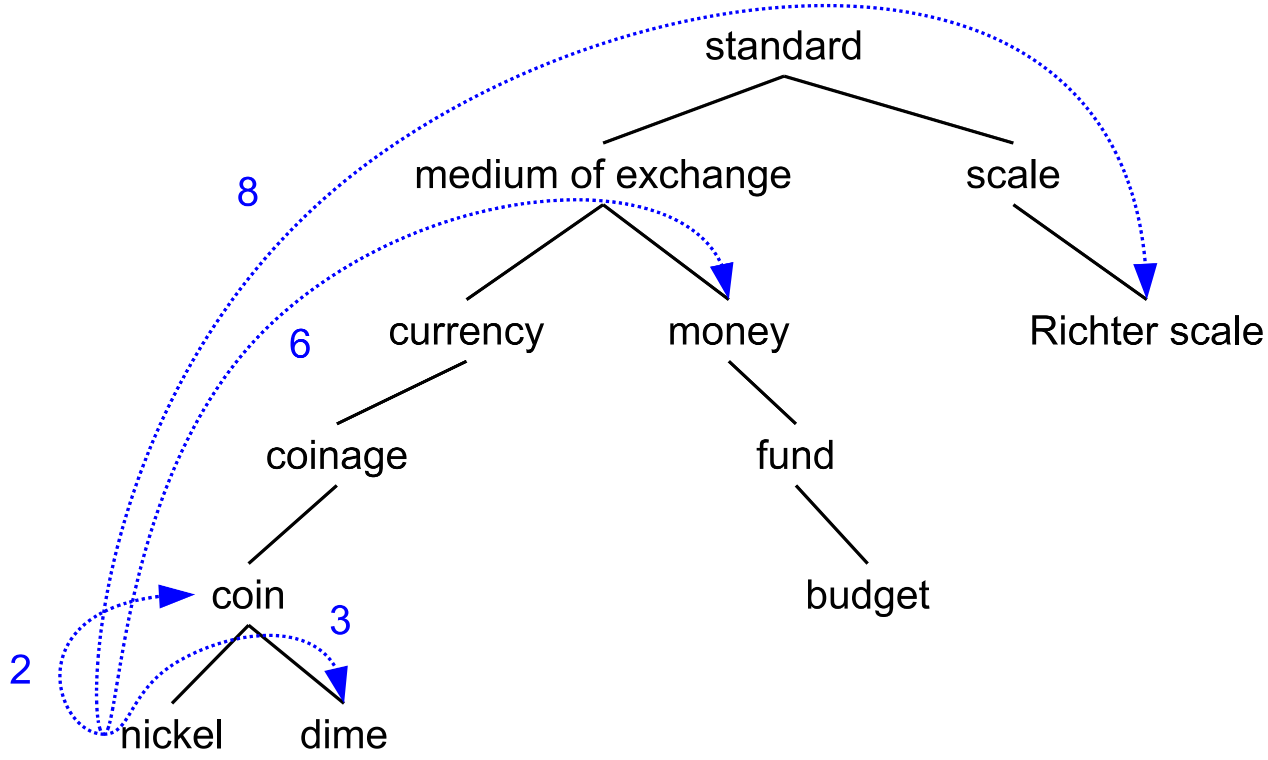
WordNet: Viewed as a graph



Two classes of similarity algorithms

- Thesaurus-based algorithms
  - Are words “nearby” in hypernym hierarchy?
  - Do words have similar glosses (definitions)?
- Distributional algorithms
  - Do words have similar distributional contexts?

Path-based similarity



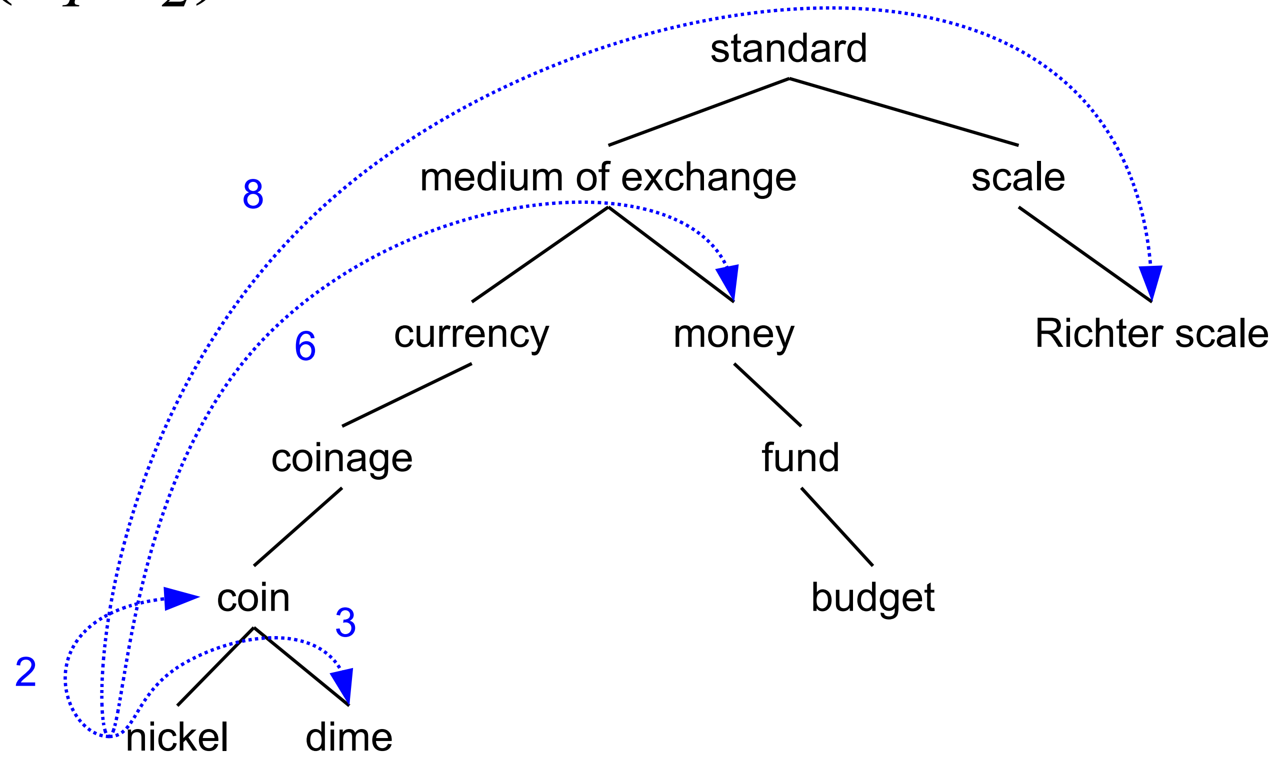
- Two concepts (senses/synsets) are similar if they are near each other in the thesaurus hierarchy
  - have a short path between them
  - concepts have path 1 to themselves

Refinements to path-based similarity

- $\text{pathlen}(c_1,c_2)$ = 1 + number of edges in the **shortest path** in the hypernym graph between sense nodes c_1 and c_2
- ranges from 0 to 1 (identity)
- $$\text{simpath}(c_1,c_2) = \frac{1}{\text{pathlen}(c_1,c_2)}$$
- $$\text{wordsim}(w_1,w_2) = \max_{c_1 \in \text{senses}(w_1), c_2 \in \text{senses}(w_2)} \text{sim}(c_1,c_2)$$

Example: path-based similarity

$$\text{simpath}(c_1,c_2) = 1/\text{pathlen}(c_1,c_2)$$



simpath(*nickel,coin*) = 1/2 = .5

simpath(*fund,budget*) = 1/2 = .5

simpath(*nickel,currency*) = 1/4 = .25

simpath(*nickel,money*) = 1/6 = .17

simpath(*nickel,standard*) = 1/6 = .17

Problem with basic path-based similarity

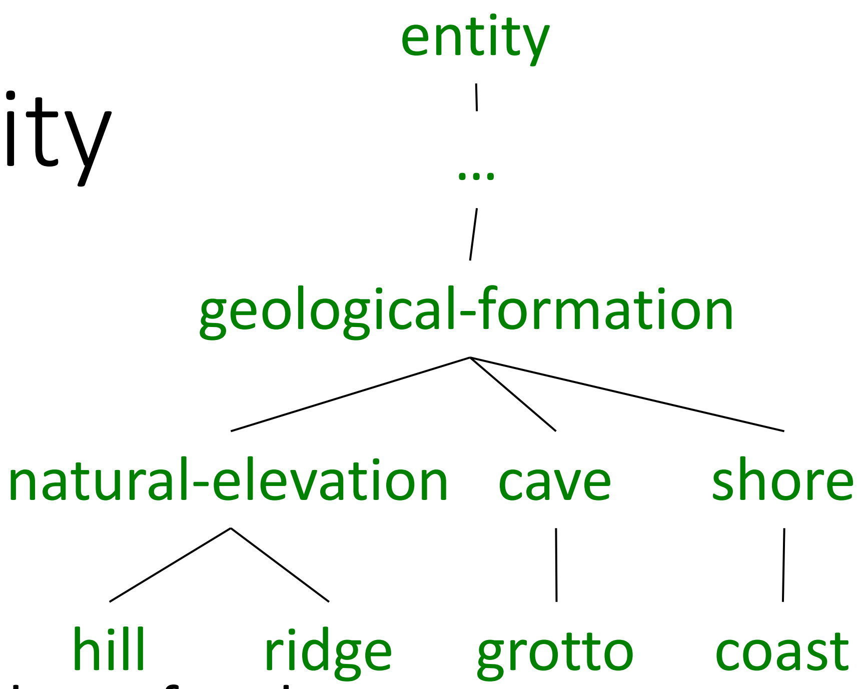
- Assumes each link represents a uniform distance
  - But *nickel* to *money* seems to us to be closer than *nickel* to *standard*
  - Nodes high in the hierarchy are very abstract
- We instead want a metric that
  - Represents the cost of each edge independently
  - Words connected only through abstract nodes
    - are less similar

Information content similarity metrics

Resnik 1995

- Let's define $P(c)$ as:
  - The probability that a randomly selected word in a corpus is an instance of concept c
  - Formally: there is a distinct random variable, ranging over words, associated with each concept in the hierarchy
    - for a given concept, each observed noun is either
      - a member of that concept with probability $P(c)$
      - not a member of that concept with probability $1\text{-}P(c)$
  - All words are members of the root node (Entity)
    - $P(\text{root})=1$ (in practice, it may not be 1)
  - The lower a node in hierarchy, the lower its probability

Information content similarity



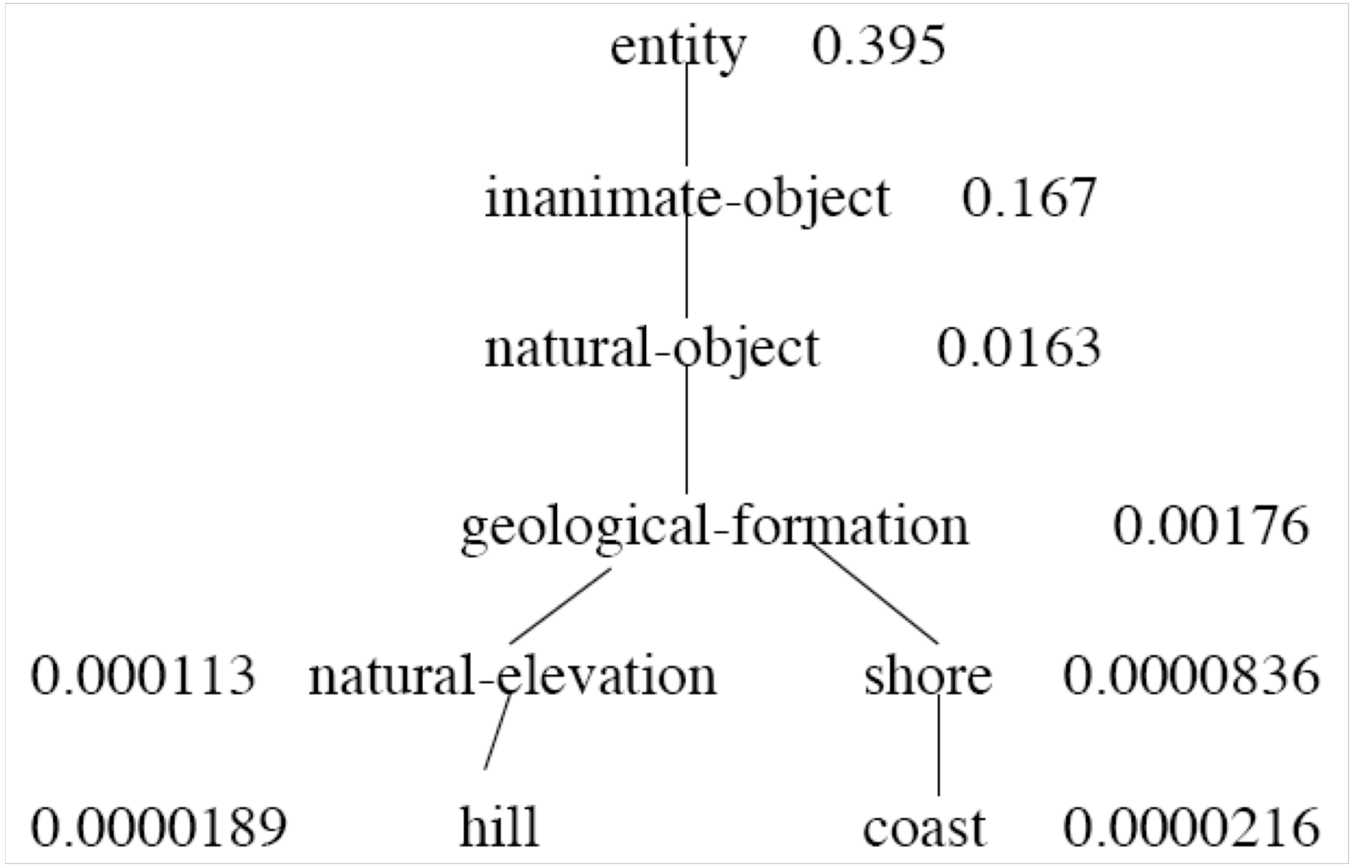
- Train by counting in a corpus
  - Each instance of `hill` counts toward frequency of *natural elevation*, *geological formation*, *entity*, etc
  - Let words(c) be the set of all words/phrases that are children of node c
    - words("geo-formation") = {hill,ridge,grotto,coast,cave,shore,natural elevation}
    - words("natural elevation") = {hill, ridge}

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

Information content similarity

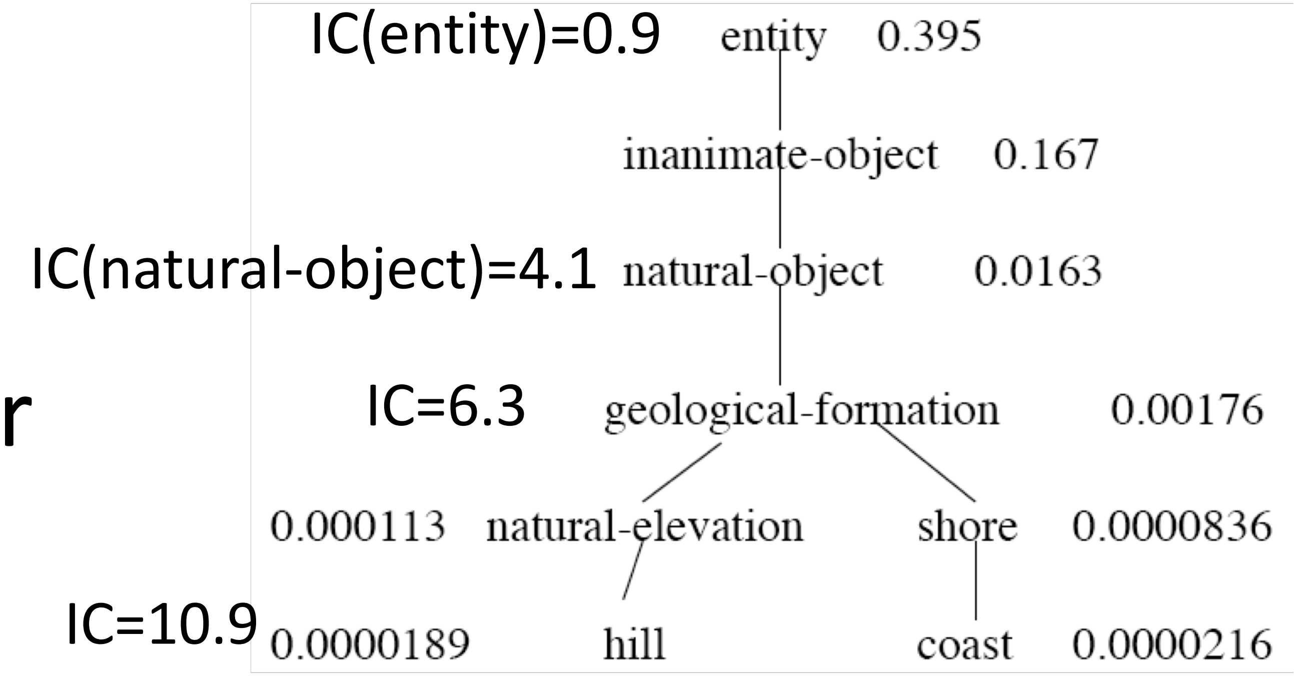
- WordNet hierarchy augmented with probabilities P(c)

D. Lin. 1998. An Information-Theoretic Definition of Similarity. ICML 1998



Information content: definitions

- Information content:
  $IC(c) = -\log P(c)$
- Most informative subsumer (Lowest common subsumer)
  $LCS(c_1,c_2) =$
  The most informative (lowest) node in the hierarchy subsuming both c_1 and c_2



Using information content for similarity: the Resnik method

Philip Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. IJCAI 1995.
Philip Resnik. 1999. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. JAIR 11, 95-130.

- The similarity between two words is related to their common information
- The more two words have in common, the more similar they are
- Resnik: measure common information as:
  - The information content of the most informative (lowest) subsumer (MIS/LCS) of the two nodes
  - $\text{sim}_{\text{resnik}}(c_1,c_2) = -\log P(\text{ LCS}(c_1,c_2)\ )$

Dekang Lin method

Dekang Lin. 1998. An Information-Theoretic Definition of Similarity. ICML

- Intuition: Similarity between A and B is not just what they have in common
- The more **differences** between A and B, the less similar they are:
  - Commonality: the more A and B have in common, the more similar they are
  - Difference: the more differences between A and B, the less similar

Dekang Lin similarity theorem

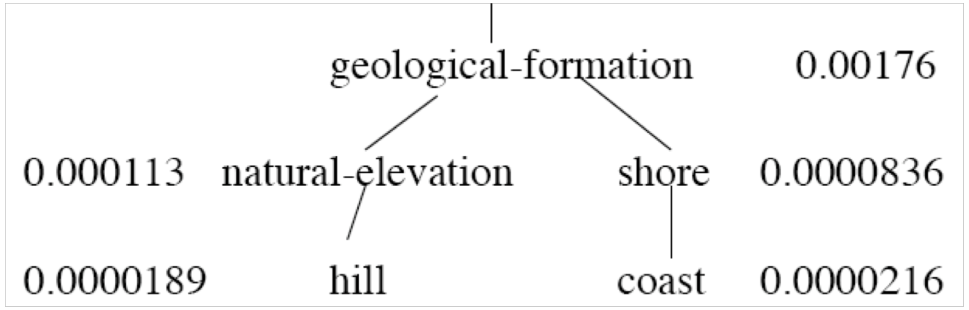
- The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are

$$sim_{Lin}(A,B) \propto \frac{IC(common(A,B))}{IC(description(A,B))}$$

- Lin (altering Resnik) defines IC(common(A,B)) as 2 x information of the LCS

$$sim_{Lin}(c_1,c_2) = \frac{2\log P(LCS(c_1,c_2))}{\log P(c_1) + \log P(c_2)}$$

Lin similarity function



$$sim_{Lin}(A,B) = \frac{2\log P(LCS(c_1,c_2))}{\log P(c_1) + \log P(c_2)}$$

$$sim_{Lin}(\text{hill},\text{coast}) = \frac{2\log P(\text{geological-formation})}{\log P(\text{hill}) + \log P(\text{coast})}$$

$$= \frac{2\ln 0.00176}{\ln 0.0000189 + \ln 0.0000216}$$

$$= .59$$

Libraries for computing thesaurus-based similarity

- NLTK
  - http://nltk.github.com/api/nltk.corpus.reader.html?highlight=similarity - nltk.corpus.reader.WordNetCorpusReader.res_similarity

- WordNet::Similarity
  - http://wn-similarity.sourceforge.net/
  - Web-based interface:
    - http://marimba.d.umn.edu/cgi-bin/similarity/similarity.cgi

Evaluating similarity

- Extrinsic (task-based, end-to-end) Evaluation:
  - Question answering
  - Spell checking
  - Essay grading
  - Word sense disambiguation
- Intrinsic Evaluation:
  - Correlation between algorithm and human word similarity ratings
    - Wordsim353: 353 noun pairs rated 0-10. *sim(plane,car)=5.77*
  - Taking multiple-choice vocabulary tests
    - `Levied is closest in meaning to:`

      `imposed, believed, requested, correlated`

Outline

- Word Senses and Word Relations
- Word Similarity
- Word Sense Disambiguation

Lexical Ambiguity

- Most words in natural languages have multiple possible meanings.
  - “pen” (noun)
    - The dog is in the pen.
    - The ink is in the pen.
  - “take” (verb)
    - Take one pill every morning.
    - Take the first right past the stoplight.

Lexical Ambiguity

- Most words in natural languages have multiple possible meanings.
  - “pen” (noun)
    - The dog is in the pen.
    - The ink is in the pen.
  - “take” (verb)
    - Take one pill every morning.
    - Take the first right past the stoplight.
- Syntax helps distinguish meanings for different parts of speech of an ambiguous word.
  - “conduct” (noun or verb)
    - John’s conduct in class is unacceptable.
    - John will conduct the orchestra on Thursday.

Motivation for Word Sense Disambiguation (WSD)

- Many tasks in natural language processing require disambiguation of ambiguous words.
  - Question Answering
  - Information Retrieval
  - Machine Translation
  - Text Mining
  - Phone Help Systems

Senses Based on Needs of Translation

- Only distinguish senses that are translate to different words in some other language.
  - play: tocar vs. jugar
  - know: conocer vs. saber
  - be: ser vs. estar
  - leave: salir vs dejar
  - take: llevar vs. tomar vs. sacar
- May still require overly fine-grained senses
  - river in French is either:
    - fleuve: flows into the ocean
    - rivière: does not flow into the ocean

Word Sense Disambiguation (WSD)

- Given
  - A word in context (*The dog is in the pen*)
  - A fixed inventory of potential word senses (*pen*^{1}*, pen*^{2})
  - Decide which sense of the word this is
- What set of senses?
  - In general: the senses in a thesaurus like WordNet
  - English-to-Spanish MT: set of Spanish translations
  - Speech Synthesis: homographs like *bass* and *bow*

Two variants of WSD task

- Lexical Sample task
  - Small pre-selected set of target words (*line, plant*)
  - And inventory of senses for each word
  - **Supervised machine learning: train a classifier for each word**
- All-words task
  - Every word in an entire text
  - A lexicon with senses for each word
  - Data sparseness: can’t train word-specific classifiers

WSD Methods

- Supervised Machine Learning
- Thesaurus/Dictionary Methods
- Semi-Supervised Learning [Leave it as your homework]

Supervised Machine Learning Approaches

- Supervised machine learning approach:
  - a training corpus of words tagged in context with their sense
  - used to train a classifier that can tag words in new text
- Summary of what we need:
  - the **tag set** (“sense inventory”)
  - the **training corpus**
  - A set of **features** extracted from the training corpus
  - A **classifier**

Supervised WSD 1: WSD Tags

- What's a tag?

  A dictionary sense?

- For example, for WordNet an instance of "bass" in a text has 8 possible tags or labels (bass1 through bass8, as noun).

8 senses of “bass” in WordNet

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

Supervised WSD 2: Get a corpus

- Lexical sample task:
  - *Line-hard-serve* corpus - 4000 examples of each
  - *Interest* corpus - 2369 sense-tagged examples
- All words:
  - **Semantic concordance**: a corpus in which each open-class word is labeled with a sense from a specific dictionary/thesaurus.
    - SemCor: 234,000 words from Brown Corpus, manually tagged with WordNet senses
    - SENSEVAL-3 competition corpora - 2081 tagged word tokens

Supervised WSD 3: Extract feature vectors

Feature vectors

- A simple representation for each observation
  (each instance of a target word)
  - **Vectors** of sets of feature/value pairs
  - Represented as an ordered list of values
  - These vectors represent, e.g., context---the window of words around the target

Lexical Ambiguity

- Most words in natural languages have multiple possible meanings.
  - “pen” (noun)
    - The dog is in the pen.
    - The ink is in the pen.
  - “take” (verb)
    - Take one pill every morning.
    - Take the first right past the stoplight.

Two kinds of features in the vectors

- **Collocational** features and **bag-of-words** features
  - **Collocational**
    - Features about words at **specific** positions near target word
      - Often limited to just word identity and POS
  - **Bag-of-words**
    - Features about words that occur anywhere in the window (regardless of position)
      - Typically limited to frequency counts

Examples

- Example text (WSJ):

  An electric guitar and **bass** player stand off to one side not really part of the scene

- Assume a window of +/- 2 from the target

Examples

- Example text (WSJ)

  An electric $\boxed{\text{guitar}}\boxed{\text{and}}$ **bass** $\boxed{\text{player}}\boxed{\text{stand}}$ off to one side not really part of the scene,

- Assume a window of +/- 2 from the target

Collocational features

- Position-specific information about the words and collocations in window
- guitar and bass player stand

$$[w_{i-2}, \mathrm{POS}_{i-2}, w_{i-1}, \mathrm{POS}_{i-1}, w_{i+1}, \mathrm{POS}_{i+1}, w_{i+2}, \mathrm{POS}_{i+2}, w_{i-2}^{i-1}, w_{i}^{i+1}]$$

[guitar, NN, and, CC, player, NN, stand, VB, and guitar, player stand]

- word 1,2,3 grams in window of ±3 is common

Bag-of-words features

- “an unordered set of words” – position ignored
- Counts of words occur within the window.
- First choose a vocabulary
- Then count how often each of those terms occurs in a given window
    - sometimes just a binary “indicator” 1 or 0

Co-Occurrence Example

- Assume we've settled on a possible vocabulary of 12 words in "bass" sentences:

[*fishing, big, sound, player, fly, rod, pound, double, runs, playing, guitar, band*]

- The vector for:

  guitar and bass player stand

  [0,0,0,1,0,0,0,0,0,0,1,0]

Syntactic Relations
(Ambiguous Verbs)

- For an ambiguous verb, it is very useful to know its direct object.
  - “played the game”
  - “played the guitar”
  - “played the risky and long-lasting card game”
  - “played the beautiful and expensive guitar”
  - “played the big brass tuba at the football game”
  - “played the game listening to the drums and the tubas”
- May also be useful to know its subject:
  - “The game was played while the band played.”
  - “The game that included a drum and a tuba was played on Friday.”

Syntactic Relations (Ambiguous Nouns)

- For an ambiguous noun, it is useful to know what verb it is an object of:
  - “**played** the piano and the horn”
  - “**wounded** by the rhinoceros’ horn”
- May also be useful to know what verb it is the subject of:
  - “the bank near the river **loaned** him $100”
  - “the bank is **eroding** and the bank has **given** the city the money to repair it”

Syntactic Relations
(Ambiguous Adjectives)

- For an ambiguous adjective, it useful to know the noun it is modifying.
  - “a brilliant young **man**”
  - “a brilliant yellow **light**”
  - “a wooden writing **desk**”
  - “a wooden acting **performance**”

Classification: definition

- *Input*:
  - a word w and some features f
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- *Output*: a predicted class $c \in C$

Classification Methods: Supervised Machine Learning

- *Input:*
  - a word *w in a text window d (which we'll call a "document")*
  - a fixed set of classes $C = \{c_1, c_2,\ldots, c_J\}$
  - A training set of *m* hand-labeled text windows again called "documents" $(d_1,y_1),\ldots,(d_m,y_m)$, y_m *is in C*
- *Output:*
  - a learned classifier $y{:}d \rightarrow c$

Classification Methods: Supervised Machine Learning

- Any kind of classifier
  - Naive Bayes
  - Logistic regression
  - Neural Networks
  - Support-vector machines
  - k-Nearest Neighbors

  - …

Applying Naive Bayes to WSD

- P(c) is the prior probability of that sense
  - Counting in a labeled training set.
- P(w|c) conditional probability of a word given a particular sense
  - $P(w|c) = count(w,c)/count(c)$
- We get both of these from a tagged corpus like SemCor

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words (context of “bass”)	Class
Training	1	fish smoked fish	f
	2	fish line	f
	3	fish haul smoked	f
	4	guitar jazz line	g
Test	5	line guitar jazz jazz	?

V = {fish, smoked, line, haul, guitar, jazz}

Priors:

$P(f) = \frac{3}{4}$

$P(g) = \frac{1}{4}$

Conditional Probabilities:

P(line|*f*) = (1+1) / (8+6) = 2/14

P(guitar|*f*) = (0+1) / (8+6) = 1/14

P(jazz|*f*) = (0+1) / (8+6) = 1/14

P(line|*g*) = (1+1) / (3+6) = 2/9

P(guitar|*g*) = (1+1) / (3+6) = 2/9

P(jazz|*g*) = (1+1) / (3+6) = 2/9

Choosing a class:

$P(f|d5) \propto 3/4 * 2/14 * (1/14)^2 * 1/14 \approx 0.00003$

$P(g|d5) \propto 1/4 * 2/9 * (2/9)^2 * 2/9 \approx 0.0006$

WSD Evaluations and baselines

- Best evaluation: **extrinsic (‘end-to-end’, `task-based’) evaluation**
  - Embed WSD algorithm in a task and see if you can do the task better!
- What we often do for convenience: **intrinsic evaluation**
  - Exact match **sense accuracy**
    - % of words tagged identically with the human-manual sense tags
  - Usually evaluate using **held-out data/test data** from same labeled corpus
- Baselines
  - Most frequent sense
  - The Lesk algorithm

Most Frequent Sense

- WordNet senses are ordered in frequency order
- So “most frequent sense” in WordNet = “take the first sense”
- Sense frequencies come from the *SemCor* corpus

Freq	Synset	Gloss
338	$plant^1$, works, industrial plant	buildings for carrying on industrial labor
207	$plant^2$, flora, plant life	a living organism lacking the power of locomotion
2	$plant^3$	something planted secretly for discovery by another
0	$plant^4$	an actor situated in the audience whose acting is rehearsed but seems spontaneous to the audience

The Simplified Lesk algorithm

- Let's disambiguate "**bank**" in this sentence:

  The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

- given the following two WordNet senses:

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

The Simplified Lesk algorithm

Choose sense with most word overlap between gloss and context
(not counting function words)

The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

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

The Corpus Lesk algorithm

- Assumes we have some sense-labeled data (like SemCor)
- Take all the sentences with the relevant word sense:

  *These short, "streamlined" meetings usually are sponsored by local **banks^1**, Chambers of Commerce, trade associations, or other civic organizations.*

- Now add these to the **gloss + examples** for each sense, call it the “signature” of a sense.
- Choose sense with most word overlap between context and signature.

Corpus Lesk: IDF weighting

- Instead of just removing function words
  - Down-weights words that occur in every “document” (gloss, example, etc)
  - These are generally function words, but is a more fine-grained measure
- Weigh each overlapping word by **inverse document frequency**

Corpus Lesk: IDF weighting

- Weigh each overlapping word by **inverse document frequency**
  - N is the total number of documents
  - df_i = "document frequency of word *i*"
  - = # of documents with word *i*

$$\mathrm{idf}_i = \log\left(\frac{N}{df_i}\right)$$

$$score(sense_i, context_j) = \sum_{w \in overlap(signature_i, context_j)} \mathrm{idf}_w$$

Semi-Supervised Learning

Problem: supervised and dictionary-based approaches require large hand-built resources

What if you don't have so much training data?

Solution: Bootstrapping

Generalize from a very small hand-labeled seed-set.

Bootstrapping

- For `bass`
  - Rely on “One sense per collocation” rule
    - A word reoccurring in collocation with the same word will almost surely have the same sense.
  - the word `play` occurs with the music sense of bass
  - the word `fish` occurs with the fish sense of bass

Sentences extracting using “fish” and “play”

We need more good teachers – right now, there are only a half a dozen who can **play** the free **bass** with ease. An electric guitar and **bass play**er stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.
The researchers said the worms spend part of their life cycle in such **fish** as Pacific salmon and striped **bass** and Pacific rockfish or snapper. And it all started when **fish**ermen decided the striped **bass** in Lake Mead were too skinny.

Summary: generating seeds

1) Hand labeling
2) "One sense per collocation":
   - A word reoccurring in collocation with the same word will almost surely have the same sense.
3) "One sense per discourse":
   - The sense of a word is highly consistent within a document - Yarowsky (1995)
   - (At least for non-function words, and especially topic-specific words)

Summary

- Word Sense Disambiguation: choosing correct sense in context
- Applications: MT, QA, etc.
- Three classes of Methods
    - Supervised Machine Learning: Naive Bayes classifier
    - Thesaurus/Dictionary Methods
    - Semi-Supervised Learning
- Main intuition
    - There is lots of information in a word's context
    - Simple algorithms based just on word counts can be surprisingly good