COMP4901K/Math4824B
Machine Learning for Natural Language Processing

Lecture 3: Introduction to NLTK

Instructor: Yangqiu Song

Today

- How shall we transform a huge text collection?
- Intro to NLTK
- (modified from Edward Loper's notes)

Corpus-based statistical approaches to tackle NLP problem

- How can a machine understand these differences?
  - Decorate the cake with the frosting
  - Decorate the cake with the kids

- Rules based approaches, i.e. hand coded syntactic constraints and preference rules:
  - The verb *decorate* require an animate being as agent
  - The object *cake* is formed by any of the following, inanimate entities (cream, dough, frosting…..)

- Such approaches have been showed to be time consuming to build, do not scale up well and are very brittle to new, unusual, metaphorical use of language
  - *To swallow* requires an animate being as agent/subject and a physical object as object
    - I swallowed his story
    - The supernova swallowed the planet

Slides Credit: Barbara Rosario

Corpus-based statistical approaches to tackle NLP problem

- A Statistical NLP approach seeks to solve these problems by automatically learning lexical and structural preferences from text collections (corpora)
- Statistical models are robust, generalize well and behave gracefully in the presence of errors and new data.
- So:
  - Get large text collections
  - Compute statistics over those collections
  - (The bigger the collections, the better the statistics)

Corpus-based statistical approaches to tackle NLP problem

- Decorate the cake with the frosting
- Decorate the cake with the kids

- From (labeled) corpora we can learn that:

  #(kids are subject/agent of decorate) > #(frosting is subject/agent of decorate)

- From (UN-labeled) corpora we can learn that:

  #(“*the kids decorate the cake*”) >> #(“*the frosting decorates the cake*”)

  #(“*cake with frosting*”) >> #(“*cake with kids*”)

  etc..

- Given these “facts” we then need a statistical model for the attachment decision

Slides Credit: Barbara Rosario

Corpus-based statistical approaches to tackle NLP problem

- Topic categorization: classify the document into semantics topics

Document 1 (sport)

The U.S. swept into the Davis Cup final on Saturday when twins Bob and Mike Bryan …

Document 2 (disasters)

One of the strangest, most relentless hurricane seasons on record reached new bizarre heights yesterday as….

- From (labeled) corpora we can learn that:

  #(sport documents containing word *Cup*) > #(disaster documents containing word *Cup*) **-- feature**

- We then need a statistical model for the topic assignment

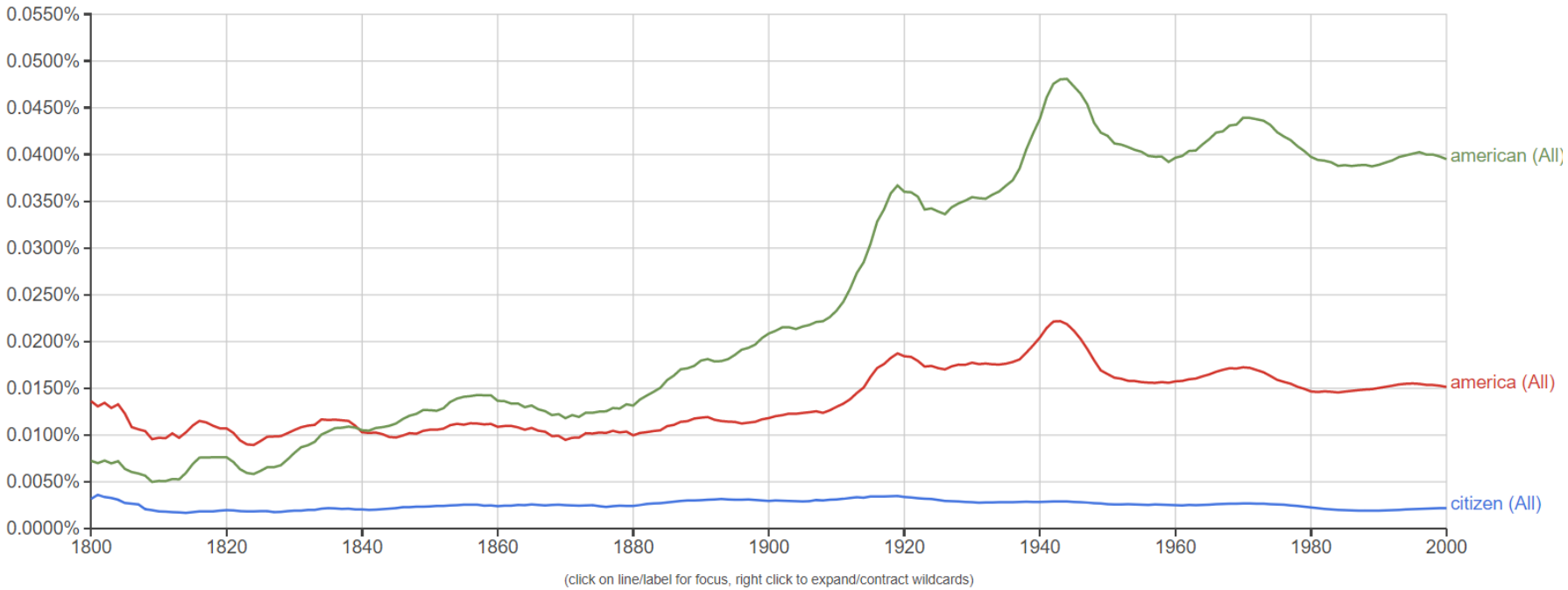
Slides Credit: Barbara Rosario

Corpus-based statistical approaches to tackle NLP problem

- Feature extractions (linguistics motivated, deep learning)
- Statistical models
- Data (corpora, labels, linguistic resources)

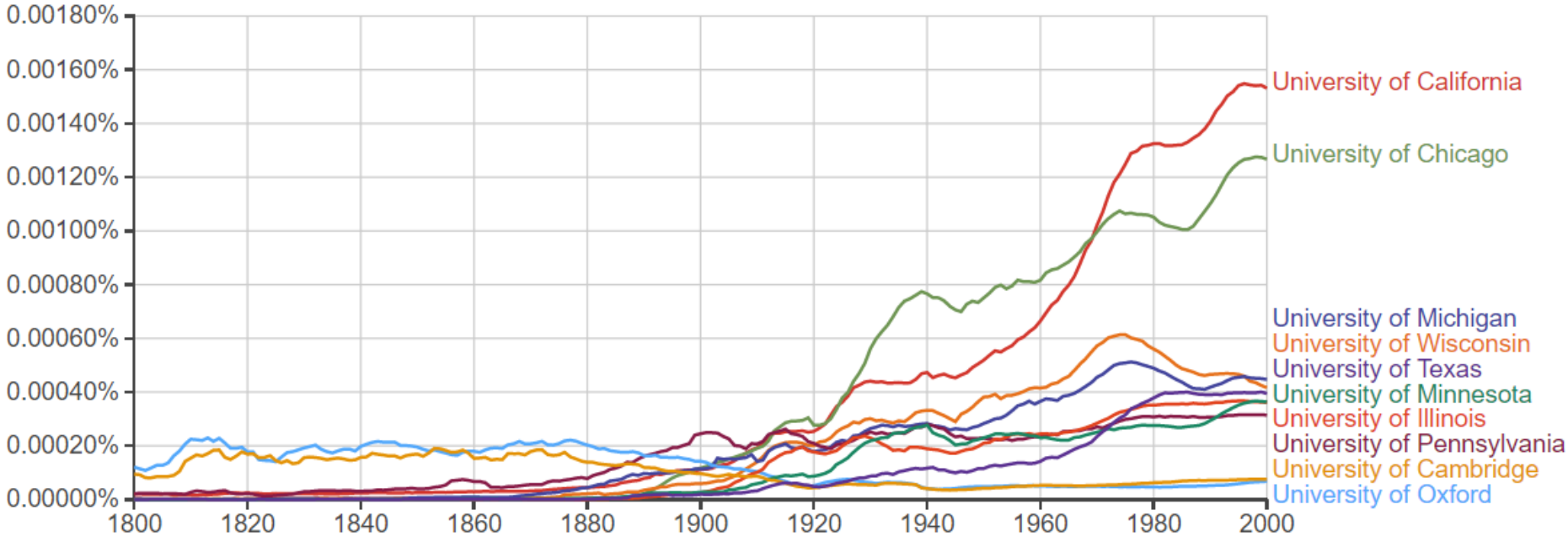
Google N-gram Dataset

- https://books.google.com/ngrams
- All counts in Google Books



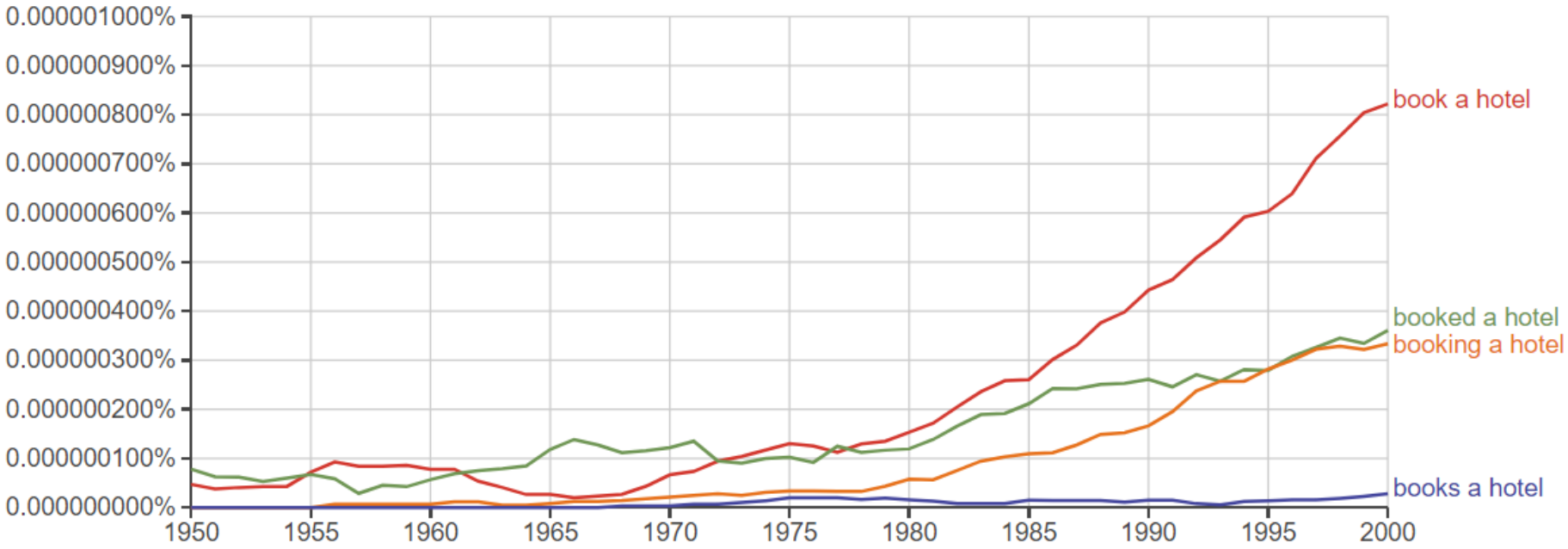
Interesting Search from Google Books

- **Wildcard search**
  - to find the most popular words following "University of", search for "University of *".



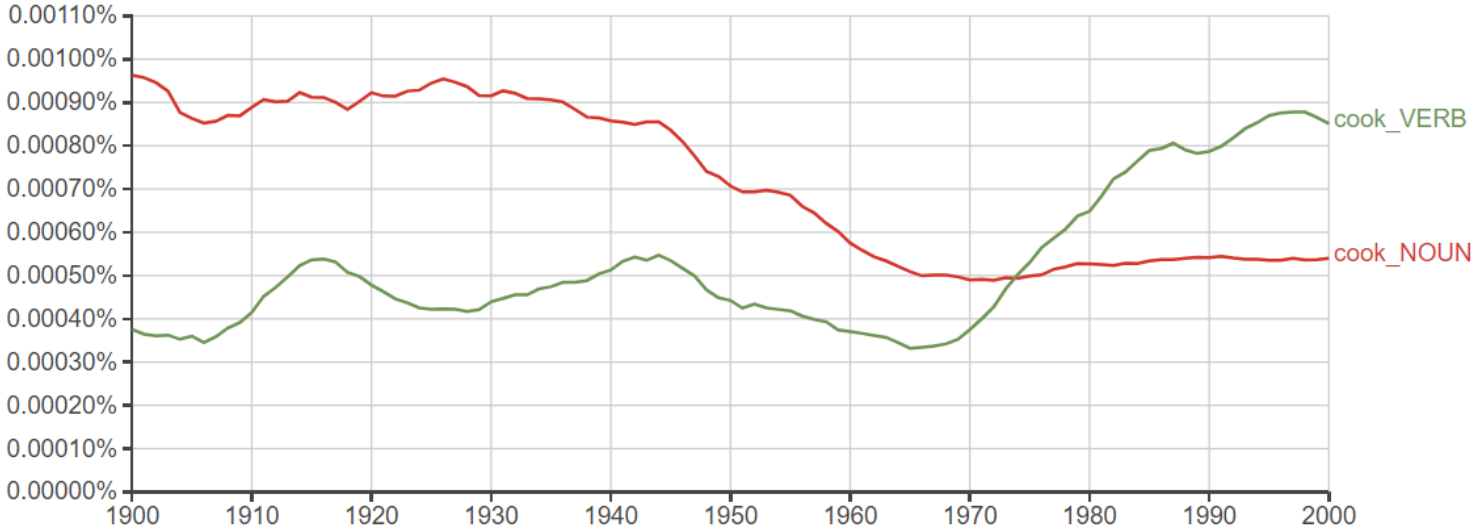
Interesting Search from Google Books

- Inflection search



Interesting Search from Google Books

- **Part-of-speech Tags**



Introduction to NLTK

- The Natural Language Toolkit (NLTK) provides:
  - Basic classes for representing data relevant to natural language processing.
  - Standard interfaces for performing tasks, such as tokenization, tagging, and parsing.
  - Standard implementations of each task, which can be combined to solve complex problems.

NLTK: Example Modules

- **`nltk.tokenize`**: processing individual elements of text, such as words or sentences.
- **`nltk.probability`**: modeling frequency distributions and probabilistic systems.
- **`nltk.tag`**: tagging tokens with supplemental information, such as parts of speech or wordnet sense tags.
- **`nltk.parse`**: high-level interface for parsing texts.
- **`nltk.app.chartparser`**: a chart-based implementation of the parser interface.
- **`nltk.chunk`**: a regular-expression based surface parser.

http://text-processing.com/demo/

NLTK: Top-Level Organization

- NLTK is organized as a flat hierarchy of packages and modules.
- Each module provides the tools necessary to address a specific task
- Modules contain two types of classes:
  - Data-oriented classes are used to represent information relevant to natural language processing.
  - Task-oriented classes encapsulate the resources and methods needed to perform a specific task.

The First Trial

- Tokens and Tokenization
- Frequency Distributions

The Tokenize Module

- It is often useful to think of a text in terms of smaller elements, such as words or sentences.
- The `nltk.tokenize` module defines classes for representing and processing these smaller elements.
- What might be other useful smaller elements?

Tokens and Types

- The term *word* can be used in two different ways:
  1. To refer to an individual occurrence of a word
  2. To refer to an abstract vocabulary item
- For example, the sentence "*my dog likes his dog*" contains five occurrences of words, but four vocabulary items.
- To avoid confusion use more precise terminology:
  1. ***Word token:*** an occurrence of a word
  2. ***Word Type:*** a vocabulary item

Text Locations

- A *text location* @ `[s:e]` specifies a region of a text:
  - `s` is the *start index*
  - `e` is the end index
- The text location @ `[s:e]` specifies the text beginning at `s`, and including everything up to (but not including) the text at `e`.
- This definition is consistent with Python *slice.*
- Think of indices as appearing *between* elements:

I	saw	a	man	
0	1	2	3	4

- Shorthand notation when location width = 1.

Text Locations (continued)

- Indices can be based on different *units*:
  - `character`
  - `word`
  - `sentence`
- Locations can be tagged with *sources* (files, other text locations – e.g., the first word of the first sentence in the file)
- Location member functions:
  - `start`
  - `end`
  - `unit`
  - `source`

Tokenization

- The simplest way to represent a **text** is with a single string.
- Difficult to process text in this format.
- Often, it is more convenient to work with a list of tokens.
- The task of converting a text from a single string to a list of tokens is known as *tokenization*.

Tokenization (continued)

- Tokenization is harder than it seems

  I'll see you in New York.

  The aluminum-export ban.
- The simplest approach is to use "graphic words" (i.e., separate words using whitespace)
- Another approach is to use regular expressions to specify which substrings are valid words.
- NLTK provides a generic tokenization interface: *`TokenizerI`*

TokenizerI

- Defines a single method, tokenize, which takes a string and returns a list of tokens
- Tokenize is independent of the level of tokenization and the implementation algorithm

Example

```
>>> import nltk, re, pprint
>>> from nltk import word_tokenize
```

```
>>> raw = """DENNIS: Listen, strange women lying in ponds
distributing swords ... is no basis for a system of
government. Supreme executive power derives from ... a mandate
from the masses, not from some farcical aquatic ceremony."""
```

```
>>> tokens = word_tokenize(raw)
>>> tokens
>>> ['DENNIS', ':', 'Listen', ',', 'strange', 'women',
'lying', 'in', 'ponds', 'distributing', 'swords', '...', 'is',
'no', 'basis', 'for', 'a', 'system', 'of', 'government', '.',
'Supreme', 'executive', 'power', 'derives', 'from', '...',
'a', 'mandate', 'from', 'the', 'masses', ',', 'not', 'from',
'some', 'farcical', 'aquatic', 'ceremony', '.']
```

Next: Corpus Statistics

- Corpus-based statistical approaches to tackle NLP problem
  - Feature extractions (linguistics motivated, deep learning)
  - Statistical models
  - Data (corpora, labels, linguistic resources)
- NLTK provides several corpora which are easy to access and use

Gutenberg Corpus

- NLTK includes a small selection of texts from the Project Gutenberg electronic text archive
  - 25,000 free electronic books
  - hosted at http://www.gutenberg.org/

```
>>> from nltk.corpus import gutenberg
>>> gutenberg.fileids()
>>> ['austen-emma.txt', 'austen-persuasion.txt', 'austen-
sense.txt', ...]
>>> emma = gutenberg.words('austen-emma.txt')
```

More detailed look

```
>>> macbeth_sentences = gutenberg.sents('shakespeare-macbeth.txt')

>>> macbeth_sentences
[['[', 'The', 'Tragedie', 'of', 'Macbeth', 'by', 'William',
'Shakespeare', '1603', ']'], ['Actus', 'Primus', '.'], ...]

>>> macbeth_sentences[1116] ['Double', ',', 'double', ',', 'toile',
'and', 'trouble', ';', 'Fire', 'burne', ',', 'and', 'Cauldron', 'bubble']

>>> longest_len = max(len(s) for s in macbeth_sentences)

>>> [s for s in macbeth_sentences if len(s) == longest_len]
[['Doubtfull', 'it', 'stood', ',', 'As', 'two', 'spent', 'Swimmers', ',',
'that', 'doe', 'cling', 'together', ',', 'And', 'choake', 'their', 'Art',
':', 'The', 'mercilesse', 'Macdonwald', ...]]
```

Other Typical Corpora

- **Web and Chat Text**
  - less formal language
  - content from a Firefox discussion forum
  - conversations overheard in New York
  - the movie script of *Pirates of the Carribean*
  - personal advertisements
  - and Amazon/Yelp reviews

Other Typical Corpora

- **Brown Corpus**
  - the first million-word electronic corpus of English
  - created in 1961 at Brown University
  - contains text from 500 sources,
  - and the sources have been categorized by genre, such as *news*, *editorial*, and so on

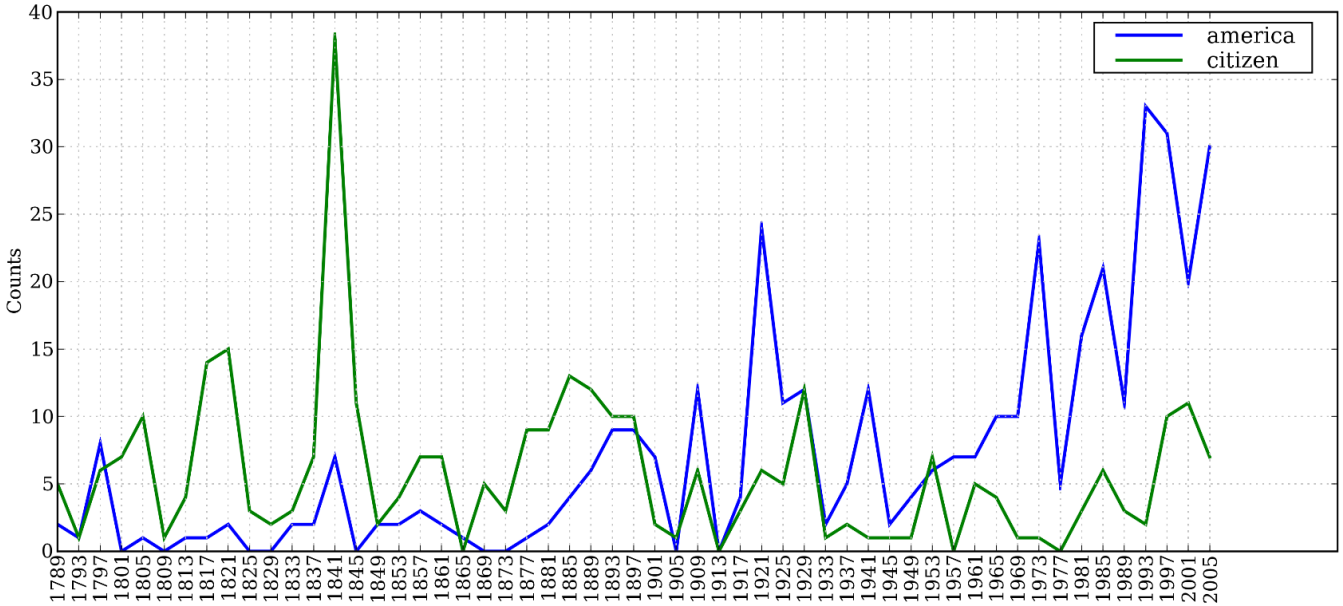
ID	File	Genre	Description
A16	ca16	news	Chicago Tribune: *Society Reportage*
B02	cb02	editorial	Christian Science Monitor: *Editorials*
C17	cc17	reviews	Time Magazine: *Reviews*
D12	cd12	religion	Underwood: *Probing the Ethics of Realtors*
E36	ce36	hobbies	Norling: *Renting a Car in Europe*
K04	ck04	fiction	W.E.B. Du Bois: *Worlds of Color*
L13	cl13	mystery	Hitchens: *Footsteps in the Night*
M01	cm01	science_fiction	Heinlein: *Stranger in a Strange Land*
N14	cn15	adventure	Field: *Rattlesnake Ridge*
P12	cp12	romance	Callaghan: *A Passion in Rome*
R06	cr06	humor	Thurber: *The Future, If Any, of Comedy*

Other Typical Corpora

- **Reuters Corpus**
  - contains 10,788 news documents totaling 1.3 million words.
  - The documents have been classified into 90 topics,
  - and grouped into two sets, called "training" and "test";
  - thus, the text with fileid 'test/14826' is a document drawn from the test set.
  - A widely used text classification benchmark dataset

Other Typical Corpora

- **Inaugural Address Corpus**
  - U.S. Presidential Inaugural Addresses



Plot of a Conditional Frequency Distribution: all words in the Inaugural Address Corpus that begin with america or citizen are counted; separate counts are kept for each address; these are plotted so that trends in usage over time can be observed; counts are not normalized for document length.

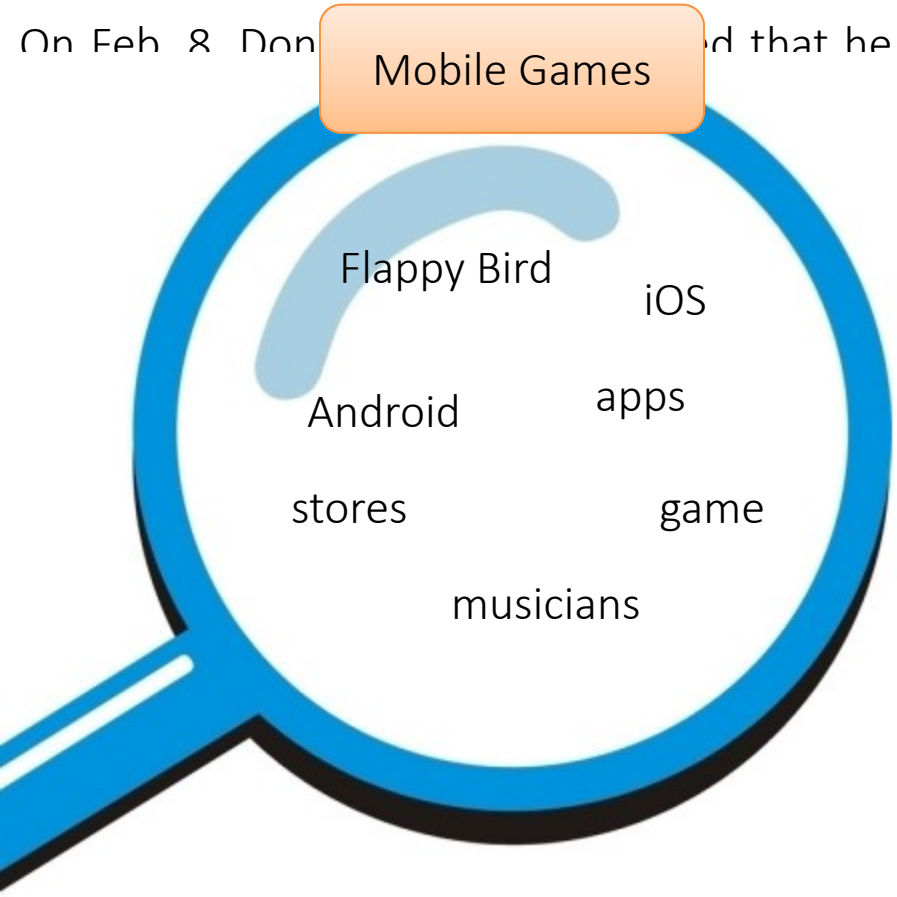
  - https://www.nltk.org/book/ch02.html

Other Typical Corpora

- **Annotated Text Corpora**
  - Many text corpora contain linguistic annotations,
    - POS tags,
    - named entities,
    - syntactic structures,
    - semantic roles,
    - and so forth.
  - convenient to access several of these corpora,
  - has data packages containing corpora and corpus samples,
  - freely downloadable for use in teaching and research.
  - http://www.nltk.org/nltk_data/

Frequency Distributions

- How can we identify the words of a text that are most informative about the topic and genre of the text?
  - You might go about finding the 50 most frequent words of a book



Counting Words Appearing in a Text

- A frequency distribution

Word Tally

the	𝍸 𝍸 𝍸 ||||
been	𝍸 𝍸 |
message	||||
persevere	|
nation	𝍸 |||

NLTK provides built-in support

```
>>> from nltk.probability import FreqDist
>>> fdist1 = FreqDist(text1) #text1 should be tokenized first
>>> print(fdist1)
<FreqDist with 19317 samples and 260819 outcomes>
>>> fdist1.most_common(50)
[(',', 18713), ('the', 13721), ('.', 6862), ('of', 6536),
('and', 6024), ('a', 4569), ('to', 4542), (';', 4072), ('in',
3916), ('that', 2982), ("'", 2684), ('-', 2552), ('his',
2459), ('it', 2209), ('I', 2124), ('s', 1739), ('is', 1695),
('he', 1661), ('with', 1659), ('was', 1632), ('as', 1620),
('"', 1478), ('all', 1462), ('for', 1414), ('this', 1280),
('!', 1269), ('at', 1231), ('by', 1137), ('but', 1113),
('not', 1103), ('--', 1070), ('him', 1058), ('from', 1052),
('be', 1030), ('on', 1005), ('so', 918), ('whale', 906),
('one', 889), ('you', 841), ('had', 767), ('have', 760),
('there', 715), ('But', 705), ('or', 697), ('were', 680),
('now', 646), ('which', 640), ('?', 637), ('me', 627),
('like', 624)]
>>> fdist1['whale']
906
```

Count word length distribution

```
>>> [len(w) for w in text1]
[1, 4, 4, 2, 6, 8, 4, 1, 9, 1, 1, 8, 2, 1, 4, 11, 5, 2, 1,
7, 6, 1, 3, 4, 5, 2, ...]

>>> fdist = FreqDist(len(w) for w in text1)

>>> print(fdist)
<FreqDist with 19 samples and 260819 outcomes>

>>> fdist
FreqDist({3: 50223, 1: 47933, 4: 42345, 2: 38513, 5:
26597, 6: 17111, 7: 14399, 8: 9966, 9: 6428, 10: 3528,
...})
```

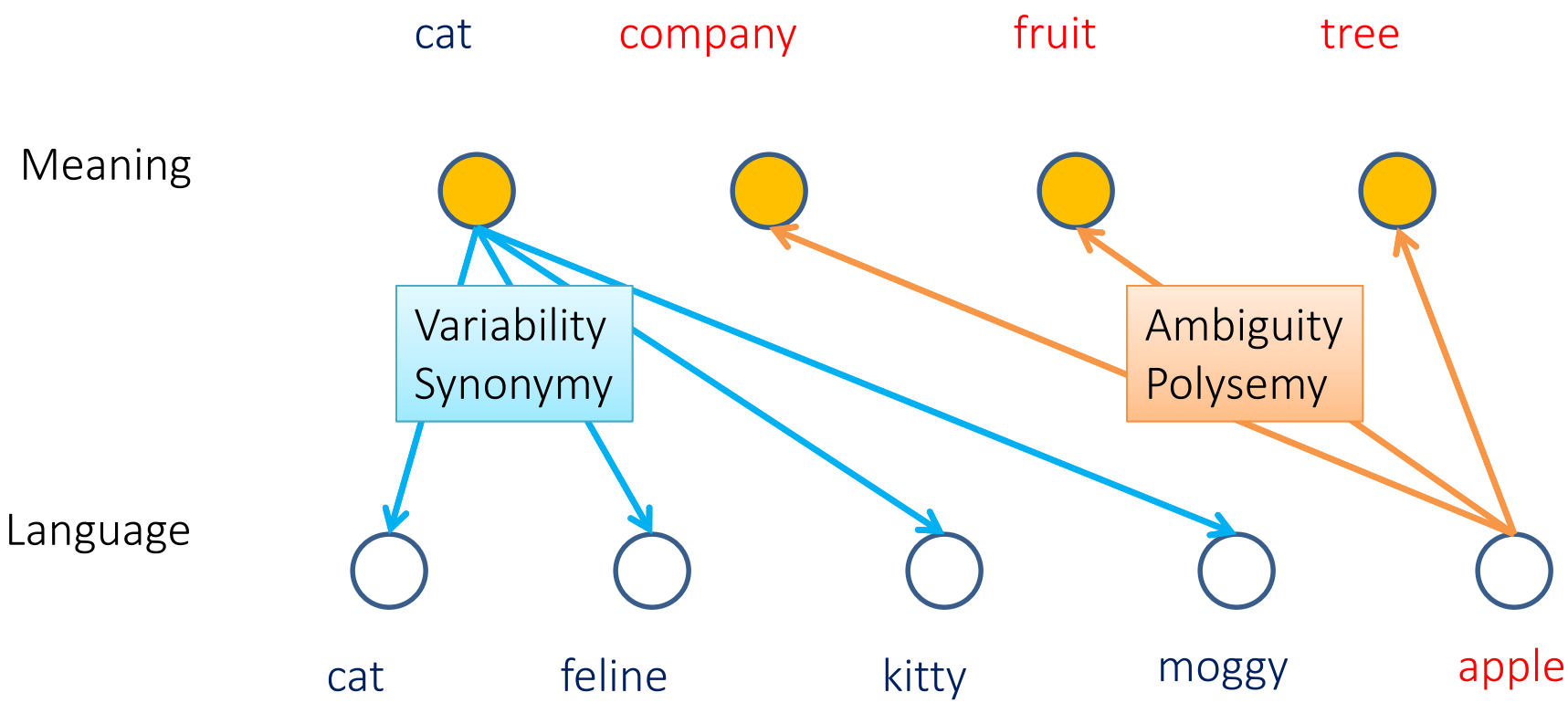
- Note that len(w) is applied for every word in text1

External Lexical/Knowledge Sources

- NLTK also provide several useful lexical/knowledge sources.
  - Wordlist Corpora: list of words such as stop words (we will introduce later)
  - A Pronouncing Dictionary: for speech generation
  - Comparative Wordlists: words in different languages
  - WordNet
    - a semantically-oriented dictionary of English,
    - similar to a traditional thesaurus but with a richer structure

Meaning of Words

- Polysemy and Synonym



Word Senses

- What does ‘bank’ mean?
  - A financial institution
    - E.g., “US bank has raised interest rates.”
  - A particular branch of a financial institution
    - E.g., “The bank on Main Street closes at 5pm.”
  - The sloping side of any hollow in the ground, especially when b ordering a river
    - E.g., “In 1927, the bank of the Mississippi flooded.”
  - A ‘repository’
    - E.g., “I donate blood to a blood bank.”

WordNet (Starting from 1985)

- A machine readable lexical database of English:
  - 117K nouns, 11K verbs, 22K adjectives, 4.5K adverbs
- Word senses grouped into synonym sets ("synsets") linked into a conceptual-semantic hierarchy
  - 82K noun synsets,
  - 13K verb synsets,
  - 18K adjectives synsets,
  - 3.6K adverb synsets
  - Avg. # of senses:
    - 1.23/noun, 2.16/verb, 1.41/adj, 1.24/adverb
- Conceptual-semantic relations
  - hypernym/hyponym

“bank”

- http://wordnet.princeton.edu/

WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: bank [Search WordNet]

Display Options: (Select option to change) [Change]

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **bank** (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"*
- S: (n) depository financial institution, **bank**, banking concern, banking company (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"*
- S: (n) **bank** (a long ridge or pile) *"a huge bank of earth"*
- S: (n) **bank** (an arrangement of similar objects in a row or in tiers) *"he operated a bank of switches"*
- S: (n) **bank** (a supply or stock held in reserve for future use (especially in emergencies))
- S: (n) **bank** (the funds held by a gambling house or the dealer in some gambling games) *"he tried to break the bank at Monte Carlo"*
- S: (n) **bank**, cant, camber (a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force)

Word to search for: cat Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **cat**, true cat (feline mammal usually having thick soft fur and no ability to roar: domestic cats; wildcats)
- S: (n) guy, **cat**, hombre, bozo, sod (an informal term for a youth or man) *"a nice guy"; "the guy's only doing it for some doll"; "the poor sod couldn't even buy a drink"*
- S: (n) **cat** (a spiteful woman gossip) *"what a cat she is!"*
- S: (n) 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 euphoric stimulant) *"in Yemen kat is used daily by 85% of adults"*
- S: (n) cat-o'-nine-tails, **cat** (a whip with nine knotted cords) *"British sailors feared the cat"*
- S: (n) Caterpillar, **cat** (a large tracked vehicle that is propelled by two endless metal belts; frequently used for moving earth in construction and farm work)
- S: (n) big cat, **cat** (any of several large cats typically able to roar and living in the wild)
- S: (n) computerized tomography, computed tomography, CT, computerized axial tomography, computed axial tomography, **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)

Verb

- S: (v) **cat** (beat with a cat-o'-nine-tails)
- S: (v) vomit, vomit up, purge, cast, sick, **cat**, be sick, disgorge, regorge, retch, puke, barf, spew, spue, chuck, upchuck, honk, regurgitate, throw up (eject the contents of the stomach through the mouth) *"After drinking too much, the students vomited"; "He purged continuously"; "The patient regurgitated the food we gave him last night"*

Lexical Relations between Nouns

- Hypernym/hyponym (between **concepts**)
  - The more general 'meal' is a hypernym of the more specific 'breakfast'
- Instance hypernym/hyponym (between **concepts** and **instances**)
  - Austen is an instance hyponym of author

Jane Austen, 1775–1817, English novelist

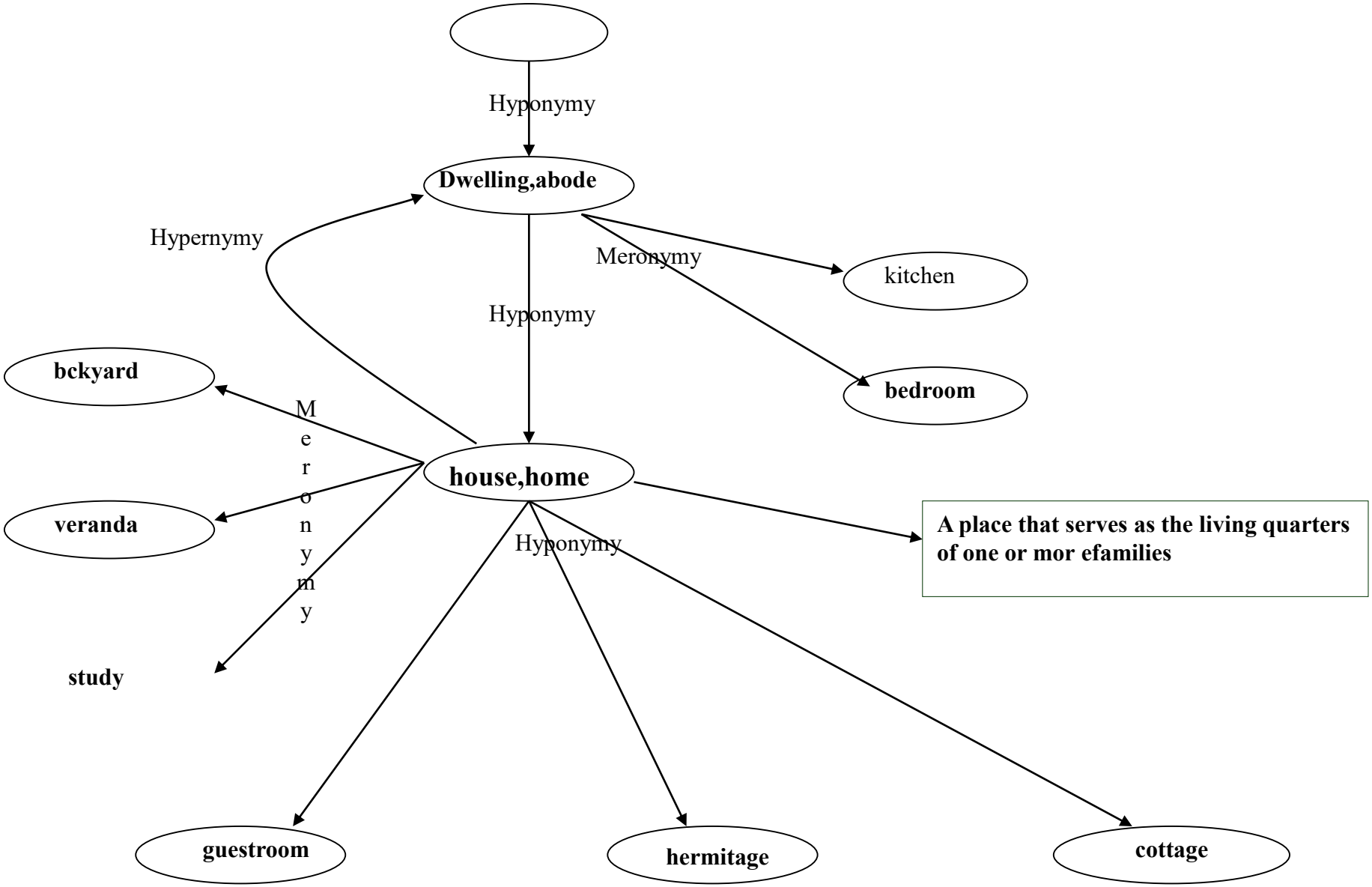
- Member holonym/meronym (**groups** and **members**)
  - professor is a member meronym of (a university's) faculty
- Part holonym/meronym (**wholes** and **parts**)
  - wheel is a part meronym of (**is a part of**) car.
- Substance meronym/holonym (**substances** and **components**)
  - flour is a substance meronym of (**is made of**) bread

Lexical Relations between Verbs

- Hypernym/troponym (between events)
  - travel/fly, walk/stroll
  - Flying is a troponym of traveling: it denotes **a specific manner of** traveling

- Entailment (between events):
  - snore/sleep
    - Snoring entails (**presupposes**) sleeping

Example of Sub-graph



Semantic Similarity/Relatedness

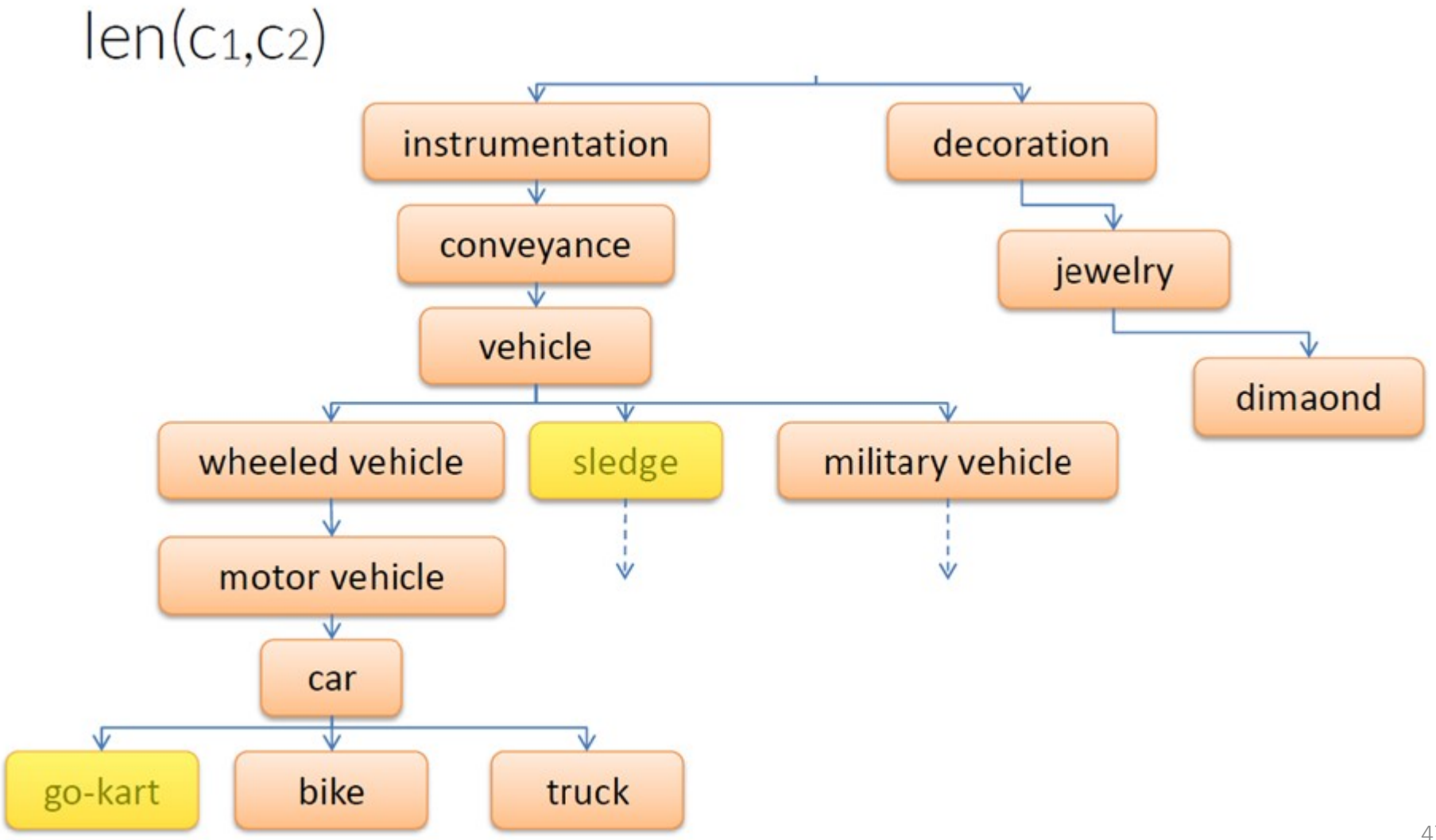
- Similarity is a specific type of relatedness: graded
  - car vs. automobile -> 1.0
  - car vs. vehicle -> 0.6
  - car vs. tire -> 0.2
  - car vs. street -> 0.1
- Similarity: **synonyms**, **hyponyms/hyperonyms**, and **siblings** are highly similar
  - doctor vs. surgeon, bike vs. bicycle
- Relatedness: **topically related** or based on any other **semantic relation**
  - heart vs. surgeon, tire vs. car

WordNet Similarity

- Path based similarity measure between words
  - **Shortest path** between two concepts (Leacock & Chodorow 1998)
    - sim = 1/|shortest path|
  - Path length to the root node from the **least common subsumer** (LCS) of the two concepts (Wu & Palmer 1994)
    - sim = 2*depth(LCS)/(depth(w_1)+depth(w_2))
  - Others

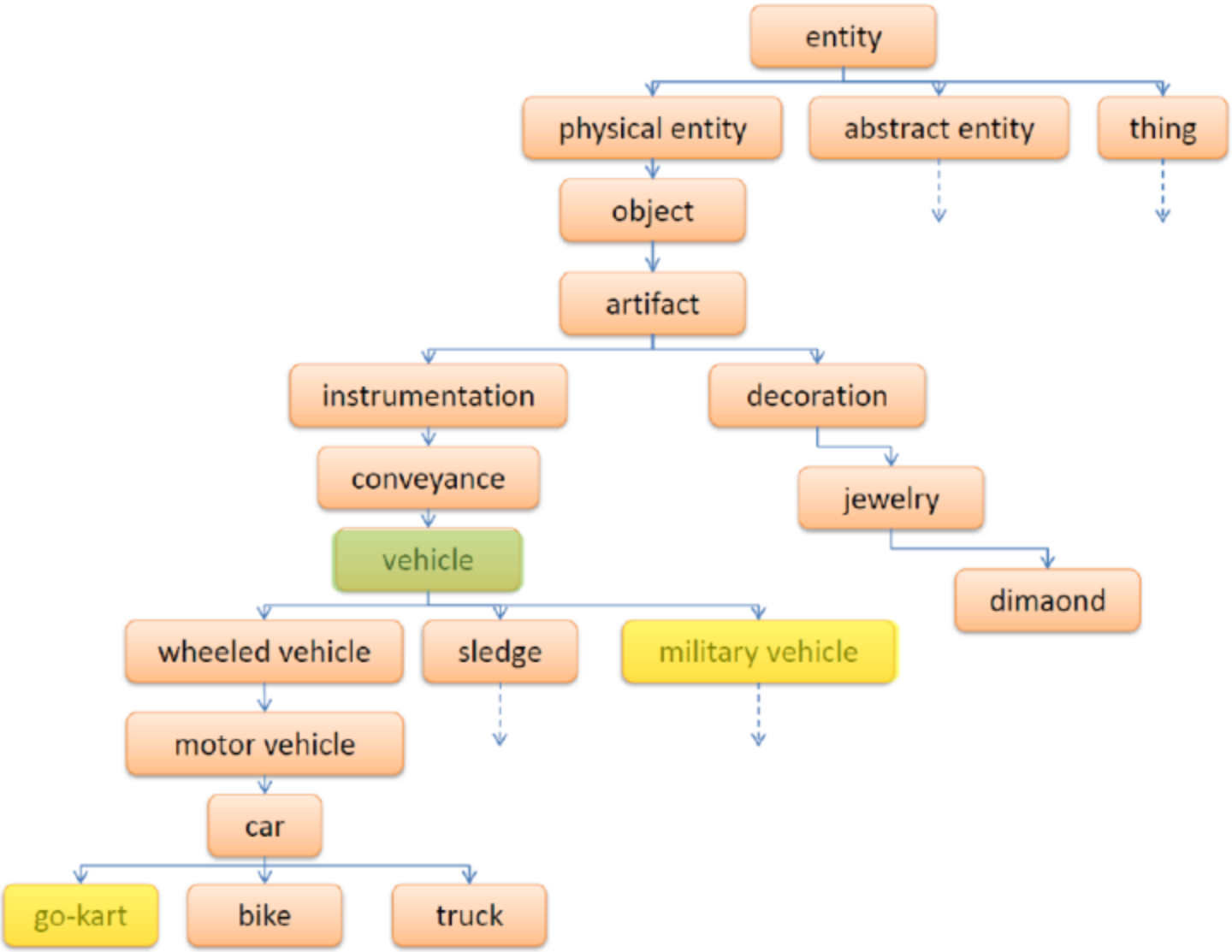
Shortest Path between Two Concepts

- sim = 1/len(c1, c2)



Least Common Subsumer

- sim = 2*depth(LCS)/(depth(w_1)+depth(w_2))



Access WordNet with NLTK

- Suppose we replace motorcar with automobile
  - a. Benz is credited with the invention of the motorcar.
  - b. Benz is credited with the invention of the automobile.
- *motorcar* and *automobile* have the same meaning, i.e. they are **synonyms**
- We can explore these words with the help of WordNet:

```
>>> from nltk.corpus import wordnet as wn
>>> wn.synsets('motorcar')
[Synset('car.n.01')]
```

- The entity car.n.01 is called a synset

```
>>> wn.synset('car.n.01').lemma_names()
['car', 'auto', 'automobile', 'machine', 'motorcar']
>>> wn.synset('car.n.01').definition()
'a motor vehicle with four wheels; usually propelled
by an internal combustion engine'
>>> wn.synset('car.n.01').examples()
['he needs a car to get to work']
```

Next

- Vector Space Model