Syntax and Parsing

Parsing

- We want to run a grammar backwards to find possible structures for a sentence
- Parsing can be viewed as a search problem
- Need to examine all structures for a string of words
- Approach:
  - Bottom-up
  - Top-down

Syntactic Structure

- To compute the syntactic structure of a sentence, must consider TWO things:
  - GRAMMAR = a formal specification of the structures allowable in a language
  - PARSING Technique = the method of analysing a sentence to determine its structure according to the grammar

Applications

- Information extraction
- Opinion Mining
- Machine translation
- Question answering

A phrase structure grammar

- $S \rightarrow NP \ \ VP$
- $VP \rightarrow V \ \ NP$
- $VP \rightarrow V \ \ NP \ \ PP$
- $NP \rightarrow NP \ \ PP$
- $NP \rightarrow N$
- $NP \rightarrow e$
- $NP \rightarrow N \ \ N$
- $PP \rightarrow P \ \ NP$

$N \rightarrow$ cats
$N \rightarrow$ claws
$N \rightarrow$ people
$N \rightarrow$ scratch
$V \rightarrow$ scratch
$P \rightarrow$ with

Phrase structure grammars
= context-free grammars (CFGs)

- $G = (T, N, S, R)$
  - T is a set of terminal symbols
  - N is a set of nonterminal symbols
  - S is the start symbol ($S \in N$)
  - R is a set of rules/productions of the form $X \rightarrow \gamma$

- A grammar G generates a language L.

Phrase structure grammars in NLP

- $G = (T, C, N, S, L, R)$
  - T is a set of terminal symbols
  - C is a set of preterminal symbols
  - N is a set of nonterminal symbols
  - S is the start symbol ($S \in N$)
  - L is the lexicon, a set of items of the form $X \rightarrow x$
    - $X \in C$ and $x \in T$
  - R is the grammar, a set of items of the form $X \rightarrow \gamma$
    - $X \in N$ and $\gamma \in (N \cup C)^*$

A phrase structure grammar

$S \rightarrow NP\ VP$

$VP \rightarrow V\ NP$

$VP \rightarrow V\ NP\ PP$

$NP \rightarrow NP\ NP$

$NP \rightarrow NP\ PP$

$NP \rightarrow N$

$NP \rightarrow e$

$PP \rightarrow P\ NP$

N → *people*

N → *fish*

N → *tanks*

N → *rods*

V → *people*

V → *fish*

V → *tanks*

P → *with*

A phrase structure grammar

S → NP VP

NP → N

NP → D N PP

NP → D N

VP → V PP

PP → P NP

N → *he*

N → *dog*

N → *eye*

V → *looked*

P → *at*

D → *the*

D → *one*

P → *with*

Probabilistic – or stochastic – context-free grammars (PCFGs)

- G = (T, N, S, R, P)
  - T is a set of terminal symbols
  - N is a set of nonterminal symbols
  - S is the start symbol ($S \in N$)
  - R is a set of rules/productions of the form $X \to \gamma$
  - P is a probability function
    - P: $R \to [0,1]$
    - $\forall X \in N, \sum_{X \to \gamma \in R} P(X \to \gamma) = 1$ $\qquad \sum_{\gamma \in T*} P(\gamma) = 1$

A PCFG

$S \rightarrow NP\ VP$	1.0
$VP \rightarrow V\ NP$	0.6
$VP \rightarrow V\ NP\ PP$	0.4
$NP \rightarrow NP\ NP$	0.1
$NP \rightarrow NP\ PP$	0.2
$NP \rightarrow N$	0.7
$PP \rightarrow P\ NP$	1.0

N → *people*	0.5
N → *fish*	0.2
N → *tanks*	0.2
N → *rods*	0.1
V → *people*	0.1
V → *fish*	0.6
V → *tanks*	0.3
P → *with*	1.0

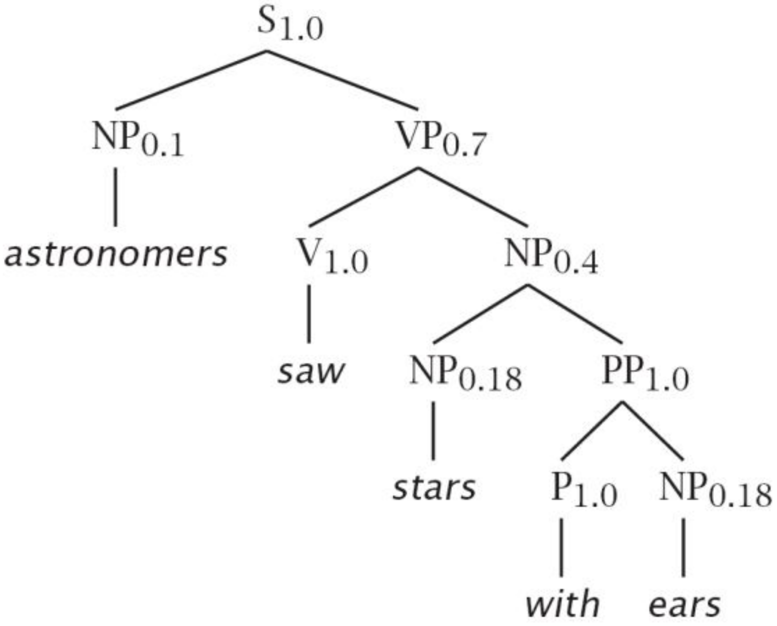
A PCFG

S → NP VP	1.0		NP → NP PP	0.4
PP → P NP	1.0		NP → *astronomers*	0.1
VP → V NP	0.7		NP → *ears*	0.18
VP → VP PP	0.3		NP → *saw*	0.04
P → *with*	1.0		NP → *stars*	0.18
V → *saw*	1.0		NP → *telescopes*	0.1

A PCFG

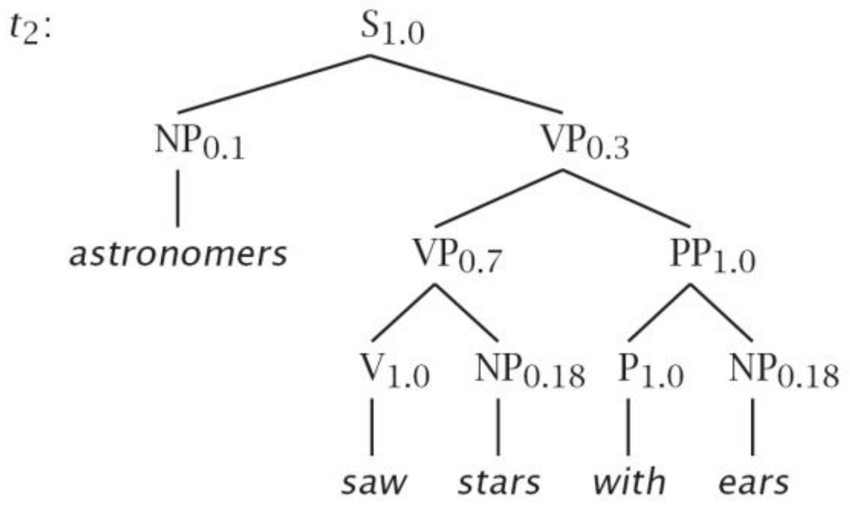
astronomers saw stars with ears

t_1:

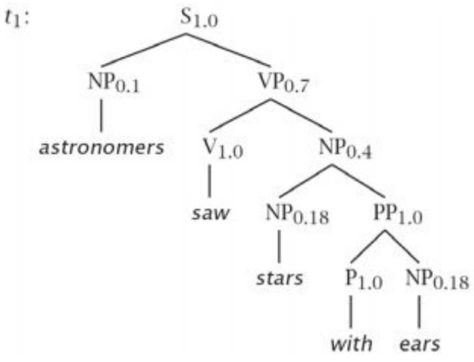


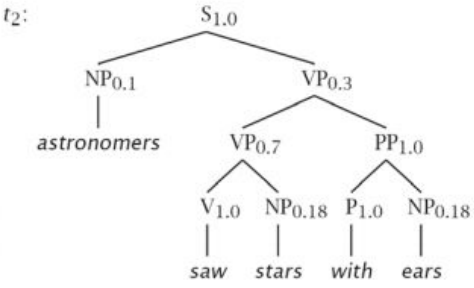
A PCFG

astronomers saw stars with ears



Probabilities





$$
\begin{aligned}
P(t_1) &= 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \\
&\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0009072 \\
P(t_2) &= 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \\
&\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0006804 \\
P(w_{15}) &= P(t_1) + P(t_2) = 0.0015876
\end{aligned}
$$

Manning, Christopher, and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

Soundness and completeness

- A parser is *sound* if every parse it returns is valid/correct
- A parser *terminates* if it is guaranteed to not go off into an infinite loop
- A parser is *complete* if for any given grammar and sentence, it is sound, produces every valid parse for that sentence, and terminates

- (For many purposes, we settle for sound but incomplete parsers: e.g., probabilistic parsers that return a *k*-best list.)

Top-down parsing

- Top-down parsing is goal directed
- A top-down parser starts with a list of constituents to be built. The top-down parser rewrites the goals in the goal list by matching one against the LHS of the grammar rules, and expanding it with the RHS, attempting to match the sentence to be derived.
- If a goal can be rewritten in several ways, then there is a choice of which rule to apply (search problem)
- Can use depth-first or breadth-first search, and goal ordering.

Bottom-up parsing

- Bottom-up parsing is data directed
- The initial goal list of a bottom-up parser is the string to be parsed. If a sequence in the goal list matches the RHS of a rule, then this sequence may be replaced by the LHS of the rule.
- Parsing is finished when the goal list contains just the start category.
- If the RHS of several rules match the goal list, then there is a choice of which rule to apply (search problem)
- Can use depth-first or breadth-first search, and goal ordering.
- The standard presentation is as *shift-reduce parsing*.

TREE Representation

- Most common method to represent how a sentence is broken into its major subparts & how these subparts are broken up in turn is using a TREE.
- Eg: Ram ate the papaya.

(S (NP (NAME Ram))   -----> LIST notation

    (VP (V ate)

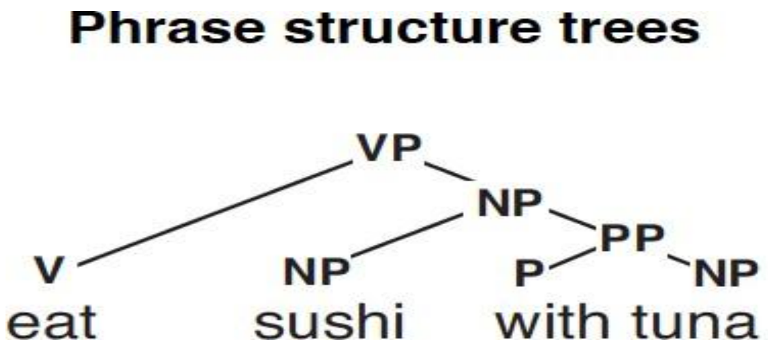
        (NP (ART the)

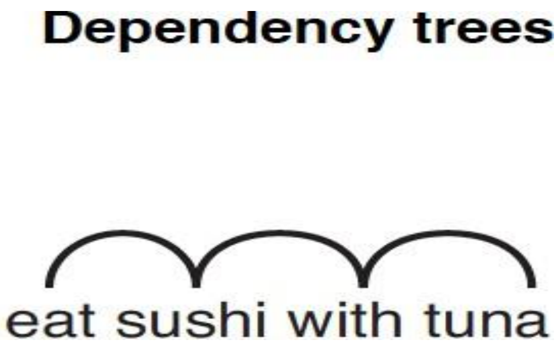
            (N papaya)))).

Tree Representation : Terminology

- Trees = a special form of GRAPH
- Structures consisting of:
  - NODES (eg. Labeled as S, NP)
  - LINKS (connecting lines/arrows)
  - ROOT (the node at the top) – (dominates all other nodes)
  - LEAVES (the nodes at the bottom)
  - “ a LINK points from a PARENT node to a CHILD node) ‘
  - Every CHILD node has a UNIQUE PARENT
  - A PARENT node may point to MANY CHILD codes
  - An ANCESTOR of a node N is defined as N’s Parent
  - A node is DOMINATED by its Ancestor node

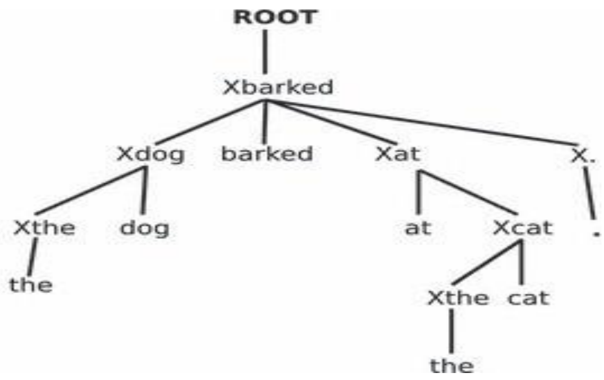
How to represent the structure





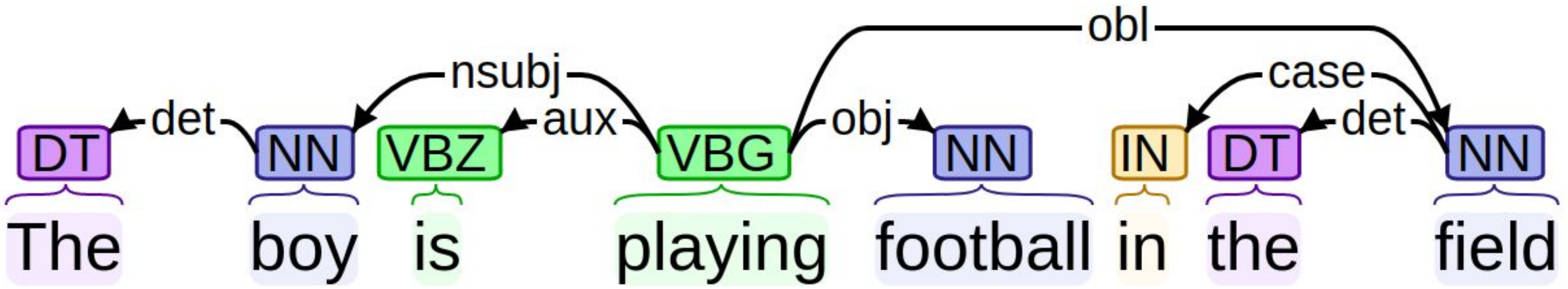
Dependency structure

- Dependency grammar describe the structure of sentences as a graph (tree)
  - Nodes represent words
  - Edges represent dependencies



Dependency structure

https://corenlp.run/



Context-free grammars

DT $\rightarrow$ {the, a}
N $\rightarrow$ {ball, garden, house, sushi }
P $\rightarrow$ {in, behind, with}
NP $\rightarrow$ **DT N**
NP $\rightarrow$ **NP PP**
PP $\rightarrow$ **P NP**

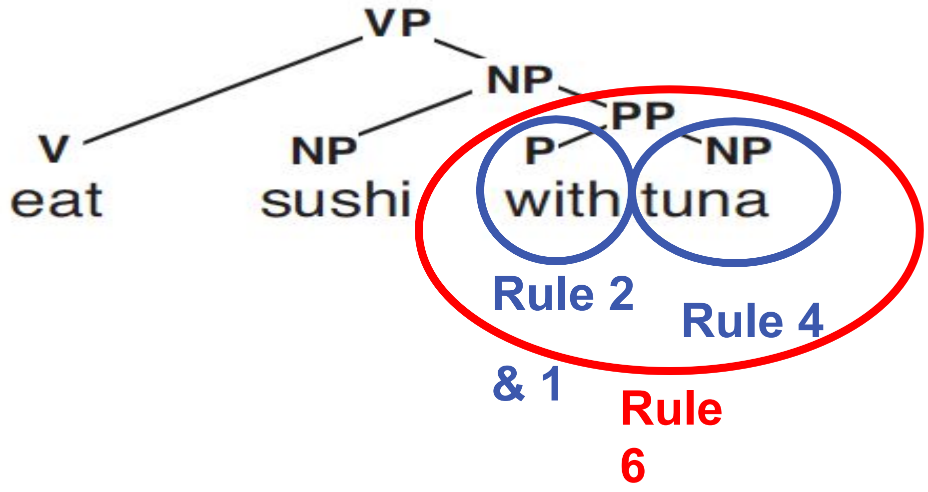
N: noun
P: preposition
NP: "noun phrase"
PP: "prepositional phrase"

PP $\longrightarrow$ P NP
PP $\longrightarrow$ P DT N
PP $\longrightarrow$ in the garden

Non-terminal: DT, N, P, NP, PP, …
Terminal: the, a, ball, garden

Parse tree defined by CFG

1 **N** → {sushi, tuna}
2 **P** → {with}
3 **V** → {eat}
4 **NP** → **N**
5 **NP** → **NP PP**
6 **PP** → **P NP**
7 **VP** → **V NP**



Simple NPs:

[He] sleeps. **(pronoun)**
[John] sleeps. **(proper name)**
[A student] sleeps. **(determiner + noun)**

Complex NPs:

[A tall student] sleeps. **(det + adj + noun)**
[The student in the back] sleeps. **(NP + PP)**
[The student who likes MTV] sleeps. **(NP + Relative Clause)**

He [eats].
He [eats sushi].
He [gives John sushi].
He [eats sushi with chopsticks].

VP → V
VP → V NP
VP → V NP PP
VP → VP PP

V → *{eats, sleeps gives,...}*

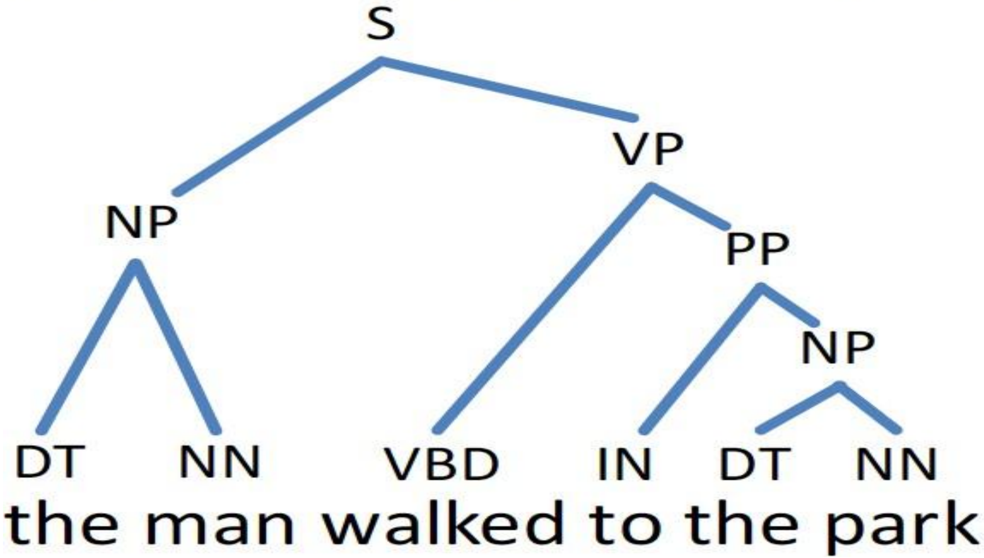
[He eats sushi].
[Sometimes, he eats sushi].
[In Japan, he eats sushi].

$S \rightarrow NP\ VP$
$S \rightarrow AdvP\ S$
$S \rightarrow PP\ S$

He says [he eats sushi].
$VP \rightarrow V_{comp}\ S$
$V_{comp} \rightarrow$ {says, think, believes}

Constituent Parse

(S (NP the man) (VP walked (PP to (NP the park))))



Key:
S = sentence
NP = noun phrase
VP = verb phrase
PP = prepositional phrase
DT = determiner
NN = noun
VBD = verb (past tense)
IN = preposition

Constituency (phrase structure)

- Phrase structure organizes words into nested constituents.
- How do we know what is a constituent? (Not that linguists don't argue about some cases.)
  - Distribution: a constituent behaves as a unit that can appear in different places:
    - John talked [to the children] [about drugs].
    - John talked [about drugs] [to the children].
    - *John talked drugs to the children about
  - Substitution/expansion/pro-forms:
    - I sat [on the box/right on top of the box/there].
  - Coordination, regular internal structure, no intrusion, fragments, semantics, …

Nonterminal in Penn

S	Sentence or clause.
SBAR	Clause introduced by a (possibly empty) subordinating conjunction.
SBARQ	Direct question introduced by a *wh*-word or *wh*-phrase.
SINV	Inverted declarative sentence.
SQ	Inverted yes/no question, or main clause of a *wh*-question.
ADJP	Adjective Phrase.
ADVP	Adverb Phrase.
CONJP	Conjunction Phrase.
FRAG	Fragment.
INTJ	Interjection.
LST	List marker. Includes surrounding punctuation.
NAC	Not A Constituent; used within an NP.
NP	Noun Phrase.
NX	Used within certain complex NPs to mark the head.

PP	Prepositional Phrase.
PRN	Parenthetical.
PRT	Particle.
QP	Quantity Phrase (i.e., complex measure/amount) within NP.
RRC	Reduced Relative Clause.
UCP	Unlike Coordinated Phrase.
VP	Verb Phrase.
WHADJP	*Wh*-adjective Phrase, as in *how hot*.
WHADVP	*Wh*-adverb Phrase.
WHNP	*Wh*-noun Phrase, e.g. *who*, *which book*, *whose daughter*, *none of which*, or *how many leopards*.
WHPP	*Wh*-prepositional Phrase, e.g., *of which* or *by whose authority*.
X	Unknown, uncertain, or unbracketable.

Pre 1990 (“Classical”) NLP Parsing

- Wrote symbolic grammar (CFG or often richer) and lexicon

S → NP VP	NN → *interest*
NP → (DT) NN	NNS → *rates*
NP → NN NNS	NNS → *raises*
NP → NNP	VBP → *interest*
VP → V NP	VBZ → *rates*

- Used grammar/proof systems to prove parses from words
- This scaled very badly and didn’t give coverage. For sentence:

Fed raises interest rates 0.5% in effort to control inflation

  - Minimal grammar: 36 parses
  - Simple 10 rule grammar: 592 parses

The rise of annotated data: The Penn Treebank

[Marcus et al. 1993, *Computational Linguistics*]

```
( (S
    (NP-SBJ (DT The) (NN move))
    (VP (VBD followed)
      (NP
        (NP (DT a) (NN round))
        (PP (IN of)
          (NP
            (NP (JJ similar) (NNS increases))
            (PP (IN by)
              (NP (JJ other) (NNS lenders)))
            (PP (IN against)
              (NP (NNP Arizona) (JJ real) (NN estate) (NNS loans))))))
      (, ,)
      (S-ADV
        (NP-SBJ (-NONE- *))
        (VP (VBG reflecting)
          (NP
            (NP (DT a) (VBG continuing) (NN decline))
            (PP-LOC (IN in)
              (NP (DT that) (NN market)))))))
    (. .)))
```

The rise of annotated data

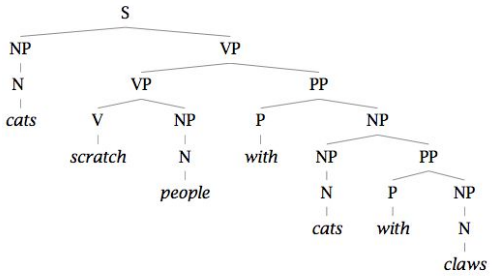
- Starting off, building a treebank seems a lot slower and less useful than building a grammar

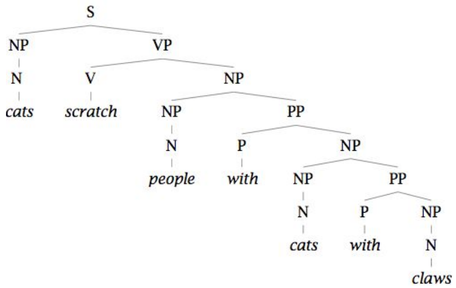
- But a treebank gives us many things
  - Reusability
    - Many parsers, POS taggers, etc.
    - Valuable resource for linguistics
  - Broad coverage
  - Frequencies and distributional information
  - A way to evaluate systems

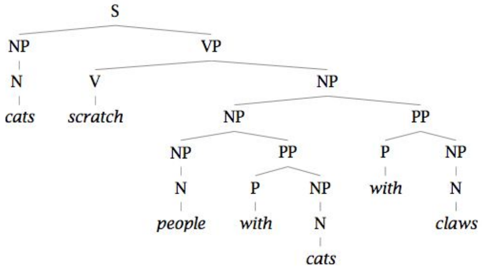
Attachment ambiguities

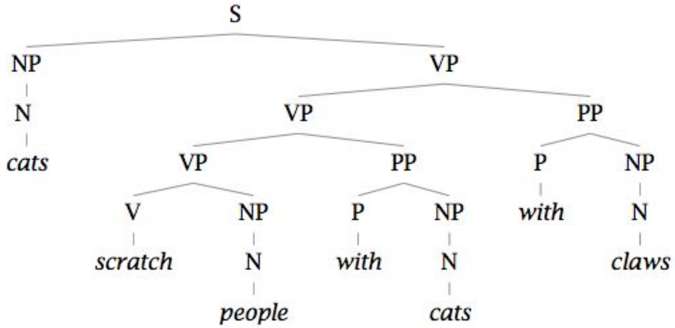
- A key parsing decision is how we ‘attach’ various constituents

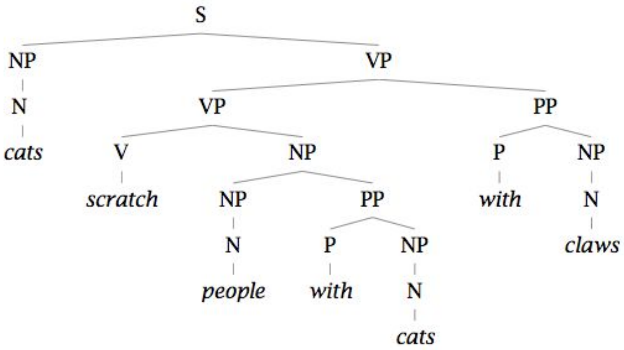
The board approved [its acquisition] [by Royal Trustco Ltd.]
[of Toronto]
[for $27 a share]
[at its monthly meeting].


S
NP VP
N VP PP
cats V NP P NP
scratch N with NP PP
people N P NP
cats with N
claws


S
NP VP
N V NP
cats scratch NP PP
N P NP
people with NP PP
N P NP
cats with N
claws


S
NP VP
N V NP
cats scratch NP PP
NP PP P NP
N P NP with N
people with N claws
cats


S
NP VP
N VP PP
cats VP PP P NP
V NP P NP with N
scratch N with N claws
people cats


S
NP VP
N VP PP
cats V NP P NP
scratch NP PP with N
N P NP claws
people with N
cats

Materials:

Stanford & University of Virginia NLP course & Jurafsky Book

Part of Speech Tagging

Introduction

POS tagging is a process that attaches each word in a sentence with a suitable tag from a given set of tags.

Introduction

- POS can analyze a word and their nearby word.
  - E.g, adjectives often followed by nouns personal pronouns often followed by verbs
  - possessive pronouns by nouns
  - Pronunciations depends on POS

Word Classes

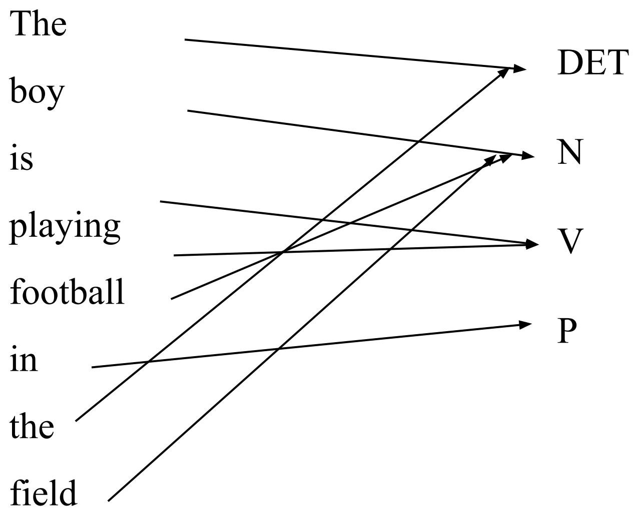
- Words that somehow ‘behave’ alike:
  - Appear in similar contexts
  - Perform similar functions in sentences
  - Undergo similar transformations
- ~9 traditional word classes of parts of speech
  - nouns, verbs, adjectives, adverbs, pronouns, prepositions, conjunctions, determiners, and exclamations

Some Examples

• N	noun	chair, bandwidth, pacing
• V	verb	study, debate, munch
• ADJ	adjective	purple, tall, ridiculous
• ADV	adverb	unfortunately, slowly
• P	preposition	of, by, to
• PRO	pronoun	I, me, mine
• DET	determiner	the, a, that, those
• CC	conjunctions	and, or
• EXC	exclamations	wow!, yahoo!

Defining POS Tagging

- The process of assigning a part-of-speech or lexical class marker to each word in a corpus:



Applications for POS Tagging

- Speech synthesis pronunciation
- Text to Speech
- Parsing:
  - e.g. *Time flies like an arrow*
  - Is *flies* an N or V?
- Word prediction in speech recognition
  - Possessive pronouns (*my, your, her*) are likely to be followed by nouns
  - Personal pronouns (*I, you, he*) are likely to be followed by verbs
- Word Sense Disambiguation
- Machine Translation

Closed vs. Open Class Words

- Closed class: relatively fixed set
  - Prepositions: of, in, by, …
  - Auxiliaries: may, can, will, had, been, …
  - Pronouns: I, you, she, mine, his, them, …
  - Usually function words (short common words which play a role in grammar)
- Open class: productive
  - English has 4: Nouns, Verbs, Adjectives, Adverbs
  - Many languages have all 4, but not all!
  - In Lakhota and possibly Chinese, what English treats as adjectives act more like verbs.

Open Class Words

- Nouns
  - Proper nouns
    - Columbia University, New York City, Arthi Ramachandran, Metropolitan Transit Center
    - English capitalizes these
    - Many have abbreviations
  - Common nouns
    - All the rest
    - German capitalizes these.

- Count nouns vs. mass nouns
  - Count: Have plurals, countable: goat/goats, one goat, two goats
  - Mass: ***Not*** countable (fish, salt, communism) (?two fishes)

- Adjectives: identify properties or qualities of nouns
  - Color, size, age, …
  - Adjective ordering restrictions in English:
    - Old blue book, ***not*** Blue old book
  - In Korean, adjectives are realized as verbs
- Adverbs: also modify things (verbs, adjectives, adverbs)
  - The very happy man walked home extremely slowly yesterday.

- Directional/locative adverbs (here, home, downhill)
- Degree adverbs (extremely, very, somewhat)
- Manner adverbs (slowly, slinkily, delicately)
- Temporal adverbs (Monday, tomorrow)

- Verbs:
  - In English, take morphological affixes (eat/eats/eaten)
  - Represent actions (walk, ate), processes (provide, see), and states (be, seem)
  - Many subclasses, e.g.
    - eats/V ⇒ eat/VB, eat/VBP, eats/VBZ, ate/VBD, eaten/VBN, eating/VBG, ...
    - Reflect morphological form & syntactic function

Closed Class Words

- Closed class words (Prep, Det, Pron, Conj, Aux, Part, Num) are generally easy to process, since we can enumerate them….but
  - Is it a Particles or a Preposition?
    - George eats up his dinner/George eats his dinner up.
    - George eats up the street/*George eats the street up.
  - Articles come in 2 flavors: definite (the) and indefinite (a, an)
    - What is this in 'this guy…'?

Tagsets

- What set of parts of speech do we use?
- Most tagsets implicitly encode fine-grained specializations of 8 basic parts of speech (POS, word classes, morphological classes, lexical tags):
  - Noun, verb, pronoun, preposition, adjective, conjunction, article, adverb
- These categories are based on morphological and distributional similarities (what words/types of words occur on the two sides of a word) and not, as you might think, semantics.
- In some cases, tagging is fairly straightforward, in other cases it is not.

Distribution of Tags

- Parts of speech follow the usual frequency-based distributional behavior
  - Most word types have only one part of speech
  - Of the rest, most have two
  - A small number of word types have lots of parts of speech
  - Unfortunately, the word types with lots of parts of speech occur with high frequency (and words that occur most frequently tend to have multiple tags)

Distribution of Tags – Brown

- To see the problem:
  - 11.5% of English words in the Brown corpus are ambiguous
  - 40% of tokens in the Brown corpus are ambiguous

Unambiguous	(1 tag)	35,340	
Ambiguous	(2-7 tags)	4,100	
	2 tags	3,760	
	3 tags	264	
	4 tags	61	
	5 tags	12	
	6 tags	2	
	7 tags	1	(“still”, “down”)

Tagset

- Adjective: stand still, the still air, still wine
- Noun: the still of the night (poetic)
- Adverb: Are you still there?
- Conjunction: It was futile, still they fought.
- Transitive Verb: Still your passions. Her fears were stilled.

Tagset

- Brown corpus tagset
- Penn Treebank tagset

Penn

CC Coordinating conjunction CD
Cardinal number
DT Determiner
EX Existential there
FW Foreign word
IN Preposition
JJ Adjective
JJR Adjective, comparative JJS
Adjective, superlative LS List item
marker
MD Modal
NN Noun, singular
NNP Proper noun, singular NNS
Noun, plural
NNPS Proper noun, plural PDT
Predeterminer
POS Posessive ending PRP
Personal pronoun PP
Possessive pronoun

RB Adverb
RBR Adverb, comparative
RBS Adverb, superlative RP
Particle
SYM Symbol TO
to
UH Interjection
VB Verb, base form VBD
Verb, past tense
VBG Verb, gerund
VBN Verb, past participle
VBP Verb, non-3rd ps. sing. present VBZ Verb, 3rd
ps. sing. present WDT wh-determiner
WP wh-pronoun
WP Possessive wh-pronoun WRB
wh-adverb

Penn

#	Pound sign
$	Dollar sign
.	Sentence-final punctuation
,	Comma
:	Colon, semi-colon
(	Left bracket character
)	Right bracket character
"	Straight double quote
`	Left open single quote
`	Left open double quote
'`	Right close single quote
"	Right close double quote

Tagset

- Using Penn Treebank tags, tag the following sentence from the Brown Corpus:
- The boy is playing football in the field

The/DT boy/NN is/VBZ playing/VBG football/NN in/IN the/DT field/NN

Tagset

Input: Plays well with others

Ambiguity: NNS/VBZ UH/JJ/NN/RB IN NNS

Output: Plays/VBZ well/RB with/IN

Tagset

- Mrs/NNP Shaefer/NNP never/RB got/VBD around/RP to/TO joining/VBG
- All/DT we/PRP gotta/VBN do/VB is/VBZ go/VB around/IN the/DT corner/NN
- Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD

Types

POS

- **Rule-based**
- **Stochastic**
- **Hybrid**

Rule-based

- Knowledge-driven taggers
- Usually rules built manually
- Regular expression using finite-state automata
- Combination of dictionary and hand written rules

Rule-based

"Change tag ***a*** to tag ***b*** when: .."

- The preceding (following) word is tagged ***z***.
- The word two before (after) is tagged ***z***.
- One of the two preceding (following) words is tagged ***z***.
- One of the three preceding (following) words is tagged ***z***.
- The preceding word is tagged ***z*** and the following word is tagged ***w***.
- The preceding (following) word is tagged ***z*** and the word two before (after) is tagged ***w***

__a,b,z__ and __w__ are part of speech tags

Stochastic

- Approaches used:
    - Word Frequency Approach
        - Most probable tag for a word in the training corpus
    - Tag Sequence Probabilities
        - Most probable sequence of tags

Stochastic

- A corpus is very important for training
- Based on the probability of tag
- Simple approach

Stochastic

	VB	TO	NN
<s>	0.19	0.0043	0.41
VB	.0038	0.045	0.34
TO	0.34	0	0.6
NN	0.56	0.00012	0.0034

Stochastic

	VB	TO	NN	PPSS
VB	0	0.0093	0	0.00012
TO	0	0	0.99	0
NN	0	0.000054	0	0.00057
PPSS	0.37	0	0	0

Computed from the 87-tag Brown corpus

Hybrid

- Combination of Rule based and Stochastic
- Use linguistic rules as well as machine learning techniques

Hybrid

Transformation-based, error-driven

- Based on rules automatically acquired

Brill, 1995
Roche,Schabes, 1995

Maximum Entropy

- Combination of several knowledge sources
- No independence is assumed
- A high number of parameters is allowed

Ratnaparkhi, 1998,
Rosenfeld, 1994
Ristad, 1997

Hybrid

From	To	If
NN	VB	previous tag is TO
to/TO conflict/NN$\rightarrow$ **VB**		
VBP	VB	one of the previous 3 tags is MD
might/MD vanish/VBP$\rightarrow$ *VB*		
NN	VB	one of the previous two tags is MD
might/MD not reply/NN$\rightarrow$ *VB*		
VB	NN	one of the previous two tags is DT
the/DT amazing play/VB$\rightarrow$ *NN*		

Hybrid

From	To	If
NN	NNS	has suffix **-s**
rules/NN→ NNS		
NN	VBN	has suffix -**ed**
tagged/NN→ VBN		
NN	VBG	has suffix **-ing**
applying/NN→ VBG		
NNS	NN	has suffix **-ss**
actress/NNS→ NN		

Links

http://www.nltk.org/book/ch05.html

https://aclanthology.org/C12-3021.pdf

Named Entity Recognition

Introduction

- Find and Classify
  - Proper Names
    - Noun Phrase
- Main Categories
  - Person
  - Location
  - Organisation
  - Date, Medical Information, Email etc...

Introduction

- University & Delhi University
- Bus & Red Bus
- Chowk & Lal Chowk
- Airport & Chennai airport
- President & Joe Biden

Introduction

- Find and Classify
  - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Named Entity Recognition (NER)

- A very important sub-task: find and classify names in text, for example:

  - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Named Entity Recognition (NER)

- A very important sub-task: find and classify names in text, for example:

  - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Person
Date
Location
Organization

Named Entity Recognition (NER)

- The uses:
  - Named entities can be indexed, linked off, etc.
  - Sentiment can be attributed to companies or products
  - A lot of IE relations are associations between named entities
  - For question answering, answers are often named entities.
- Concretely:
  - Many web pages tag various entities, with links to bio or topic pages, etc.
    - Reuters' OpenCalais, Evri, AlchemyAPI, Yahoo's Term Extraction, …
  - Apple/Google/Microsoft/… smart recognizers for document content

The Named Entity Recognition Task

Task: Predict entities in a text

Foreign	ORG
Ministry	ORG
spokesman	O
Shen	PER
Guofang	PER
told	O
Reuters	ORG
:	:

Standard evaluation is per entity, *not* per token

The ML sequence model approach to NER

Training

1. Collect a set of representative training documents
2. Label each token for its entity class or other (O)
3. Design feature extractors appropriate to the text and classes
4. Train a sequence classifier to predict the labels from the data

Testing

1. Receive a set of testing documents
2. Run sequence model inference to label each token
3. Appropriately output the recognized entities

Encoding classes for sequence labeling

	IO encoding	IOB/BIO encoding
Ram	PER	B-PER
showed	O	O
Krish	PER	B-PER
Da	PER	B-PER
Vinci	PER	I-PER
‘s	O	O
new	O	O
painting	O	O

Rule Based NER (1)

- **Create regular expressions to extract:**
  - Telephone number
  - E-mail
  - Capitalized names

Rule Based NER (1)

- Regular expressions provide a flexible way to match strings of text, such as particular characters, words, or patterns of characters

Suppose you are looking for a word that:

1. starts with a capital letter “P”
2. is the first word on a line
3. the second letter is a lower case letter
4. is exactly three letters long
5. the third letter is a vowel

the regular expression would be “^P[a-z][aeiou]” where

^ - indicates the beginning of the string

[a-z] – any letter in range a to z

[aeiou] – any vowel

Rule Based NER (1)

- **Create regular expressions to extract:**
  - Telephone number
  - E-mail
  - Capitalized names

  blocks of digits separated by hyphens

  RegEx = (\d+\-)+\d+

Rule Based NER (1)

- **Create regular expressions to extract:**
  - Telephone number
  - E-mail
  - Capitalized names

  blocks of digits separated by hyphens

  RegEx = (\d+\-)+\d+

  - matches valid phone numbers like 900-865-1125 and 725-1234
  - incorrectly extracts social security numbers 123-45-6789
  - fails to identify numbers like 800.865.1125 and (800)865-CARE

  Improved RegEx = (\d{3}[-.\ ()]){1,2}[\dA-Z]{4}

Rule Based NER (2)

- **Create rules to extract locations**
  - Capitalized word + {city, center, river} indicates location

    Ex. *New York city*

    Hudson river

  - Capitalized word + {street, boulevard, avenue} indicates location

    Ex. *Fifth avenue*

Rule Based NER (3)

- **Use context patterns**
  - [*PERSON*] earned [*MONEY*]
    Ex. *Frank earned $20*
  - [*PERSON*] joined [*ORGANIZATION*]
    Ex. *Sam joined IBM*
  - [*PERSON*],[*JOBTITLE*]
    Ex. *Mary, the teacher*

  still not so simple:
  - [*PERSON|ORGANIZATION*] fly to [*LOCATION|PERSON|EVENT*]
    Ex. *Jerry flew to Japan*
    Sarah flies to the party
    Delta flies to Europe

Rule Based NER (3)

- **Use context patterns**

  still not so simple:

  – [*PERSON*|*ORGANIZATION*|*ANIMAL*] fly to [*LOCATION*|*PERSON*|*EVENT*]

    Ex. *Jerry flew to Japan*

    Sarah flies to the party

    Delta flies to Europe

    bird flies to trees

    bee flies to the wood

Why simple things would not work?

- Capitalization is a strong indicator for capturing proper names, but it can be tricky:
  - first word of a sentence is capitalized
  - sometimes titles in web pages are all capitalized
  - all nouns in German are capitalized

Why simple things would not work?

- We already have discussed that currently no gazetteer contains all existing proper names.
- New proper names constantly emerge

movie titles

books

singers

restaurants

etc.

Learning System

- *Supervised* learning
  - Labeled training examples
  - k-Nearest Neighbors, Decision Trees,SVM, HMM…
  - example: NE recognition, POS tagging, Parsing, Information extraction
- *Unsupervised* learning
  - labels must be automatically discovered
  - method: clustering
  - example: NE disambiguation, text classification

Learning System

- *Semi-supervised* learning
  - small percentage of training examples are labeled, the rest is unlabeled
  - methods: bootstrapping, active learning, co-training, self-training
  - example: NE recognition, POS tagging, Parsing, …

Machine Learning NER

Adam_B Smith_I works_O for_O IBM_B ,_O London_B ._O

- **NED**: Identify named entities using BIO tags
  - B beginning of an entity
  - I continues/inside the entity
  - O word outside the entity

Machine Learning NER

Adam_B-PER Smith_I-PER works_O for_O **IBM_B-ORG** ,_O **London_B-LOC** ._O

- **NED**: Identify named entities using BIO tags
  - B beginning of an entity
  - I continues the entity
  - O word outside the entity
- **NEC**: Classify into a predefined set of categories
  - Person names
  - Organizations (companies, governmental organizations, etc.)
  - Locations (cities, countries, etc.)
  - Miscellaneous (movie titles, sport events, etc.)

Materials:

Stanford

Semantic

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
    - banks ---> bank
    - ran ---> run

Lemmas have senses

- Bank
  - `Sense 1`
  - `Sense 2`

Homonymy

Homonyms: words that share a same form(spelling or pronunciation) but have unrelated, distinct meanings:

Homographs (bank/bank, bat/bat)

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

Homophones:

1. Write and right
2. Piece and peace

Homonymy

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

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 2: "A financial institution"
  - Sense 1: "The building belonging to a financial institution"
- The ***key*** broke in the lock.
- The ***key*** problem was not one of quality but of quantity.
- A **polysemous** word has **related** meanings

Metonymy or Systematic Polysemy

- 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?

- Two senses of `serve`?
  - `Which flights` **`serve`** `breakfast?`
  - `Does Lufthansa` **`serve`** `Chennai?`
  - ?Does Lufthansa serve breakfast and Chennai?

Synonyms

- Word that have the same meaning in some or all contexts.
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - Water / H_20
- Two lexemes are synonyms
  - if they can be substituted for each other in all situations

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/$H_2 0$
  - Big/large
  - Brave/courageous

Synonymy

- 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`

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/hyper	vehicle	fruit	furniture
Subordinate/hyponym	car	mango	chair

Hyponyms and Instances

- WordNet 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...`

Applications

- Information Extraction
- Information Retrieval
- Question Answering
- Bioinformatics and Medical Informatics
- Machine Translation

WordNet 3.0

- A hierarchically organized lexical database
- On-line thesaurus + aspects of a dictionary
    - Some other languages available or under development
        - (Arabic, Finnish, German, Portuguese…)

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

WordNet 3.0

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

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} → *meal*^{1}
Hyponym	Subordinate	From concepts to subtypes	*meal*^{1} → *lunch*^{1}
Member Meronym	Has-Member	From groups to their members	*faculty*^{2} → *professor*^{1}
Has-Instance		From concepts to instances of the concept	*composer*^{1} → *Bach*^{1}
Instance		From instances to their concepts	*Austen*^{1} → *author*^{1}
Member Holonym	Member-Of	From members to their groups	*copilot*^{1} → *crew*^{1}
Part Meronym	Has-Part	From wholes to parts	*table*^{2} → *leg*^{3}
Part Holonym	Part-Of	From parts to wholes	*course*^{7} → *meal*^{1}
Antonym		Opposites	*leader*^{1} → *follower*^{1}

Word Similarity

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

Word Similarity

- Information retrieval
- Question answering
- Machine translation
- Natural language generation
- Language modeling
- Automatic essay grading
- Plagiarism detection
- Document clustering

Word similarity and word relatedness

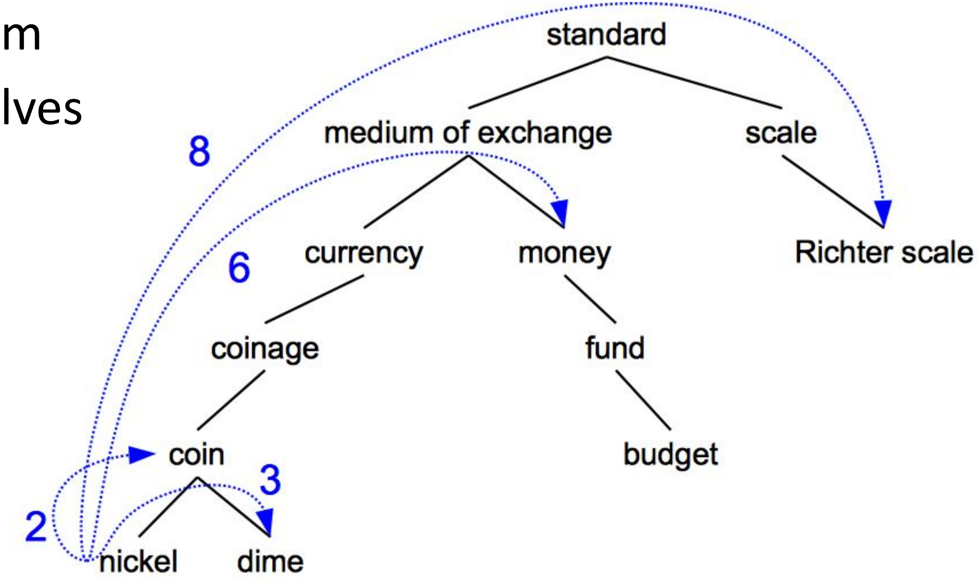
- We often distinguish **word similarity** from **word relatedness**
  - **Similar words**: near-synonyms
  - **Related words**: can be related any way
    - `car, bicycle:` **similar**
    - `car, gasoline:` **related**, not similar

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)

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

Example: path-based similarity

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

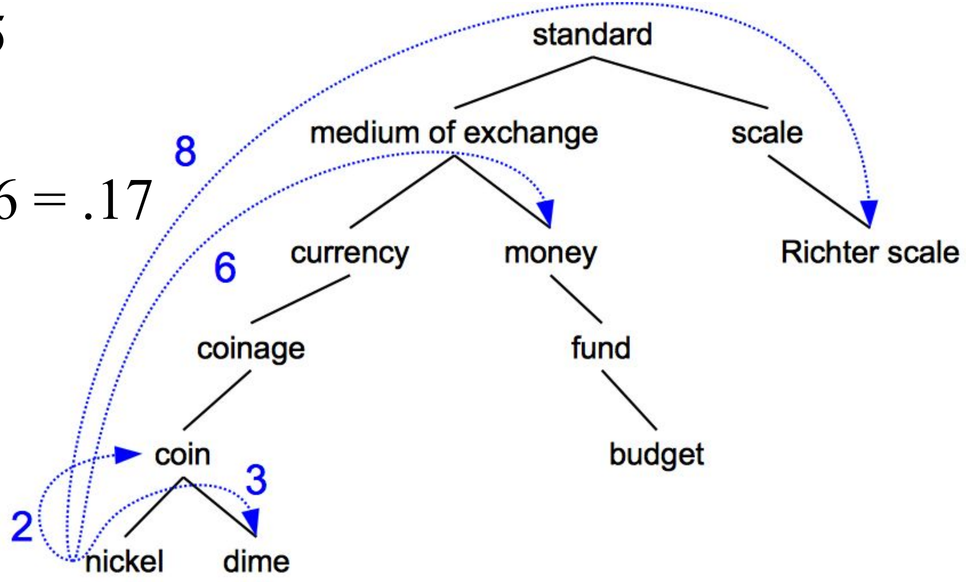
simpath(*nickel,coin*) = 1/2 = .5

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

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

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

simpath(*coinage,Richter scale*) = 1/6 = .17



Problem with basic path-based similarity

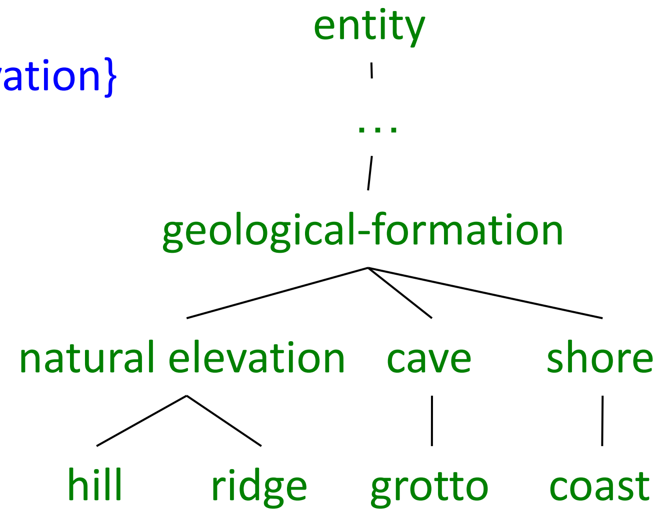
- Assumes each link represents a uniform distance
  - But *nickel* to *money* seems to us to be closer than *nickel* to *standard*
  - Nodes high in the hierarchy are very abstract
- We instead want a metric that
  - Represents the cost of each edge independently
  - Words connected only through abstract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

Information content similarity

- 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 that are children of node c + c
    - words("geo-formation") = {hill,ridge,grotto,coast,cave,shore,natural elevation, geo-formation}
    - words("natural elevation") = {hill, ridge, natural elevation}

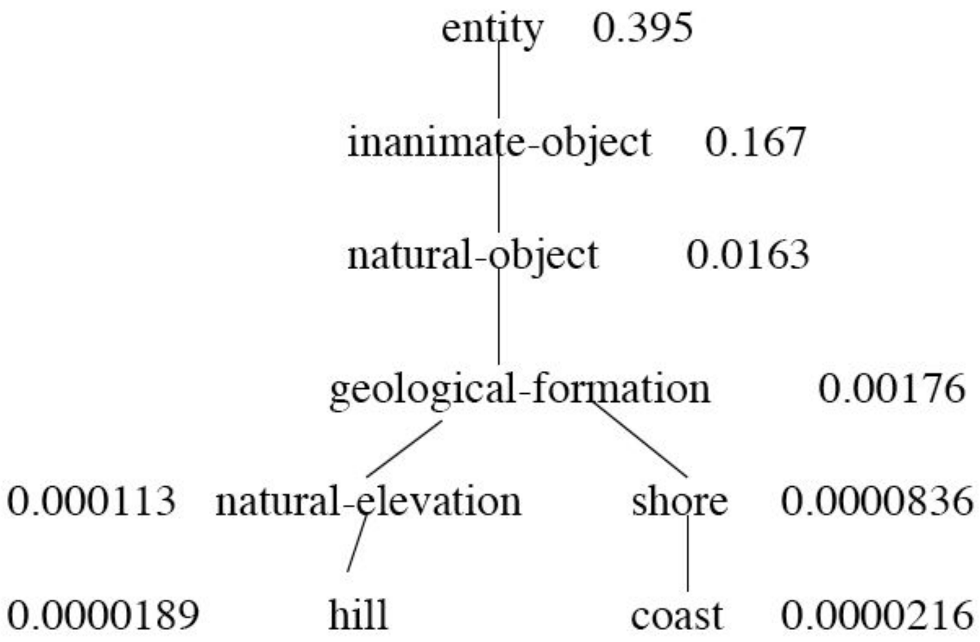
$$P(c)=\frac{\sum_{w\in words(c)} count(w)}{N}$$



Information content similarity

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

- WordNet hierarchy augmented with probabilities P(c)



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
- Commonality: IC(common(A,B))
- Difference: IC(description(A,B)-IC(common(A,B))

Dekang Lin method

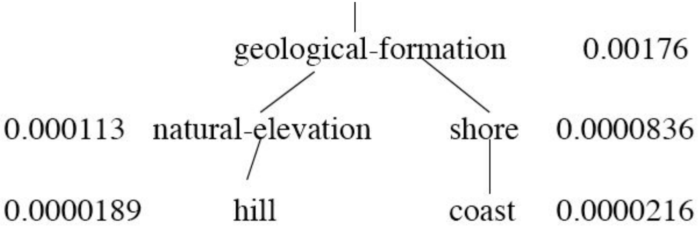
- The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are

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

- Lin 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



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

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

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

$$= .59$$

Distributional word similarity

- Example:

```
A bottle of tesgüino is on the table
Everybody likes tesgüino
Tesgüino makes you drunk
We make tesgüino out of corn.
```

- From context words humans can guess ***tesgüino*** means
  - an alcoholic beverage like **beer**
- Intuition for algorithm:
  - Two words are similar if they have similar word contexts.

Term-document matrix

- Two documents are similar if their vectors are similar

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

The Term-Context matrix

- Instead of using entire documents, use smaller contexts
  - Paragraph
  - Window of 10 words
- A word is now defined by a vector over counts of context words

Sample contexts: 20 words (Brown corpus)

- equal amount of sugar, a sliced lemon, a tablespoonful of **apricot** preserve or jam, a pinch each of clove and nutmeg,
- on board for their enjoyment. Cautiously she sampled her first **pineapple** and another fruit whose taste she likened to that of
- of a recursive type well suited to programming on the **digital** computer. In finding the optimal R-stage policy from that of
- substantially affect commerce, for the purpose of gathering data and **information** necessary for the study authorized in the first section of this

Term-context matrix for word similarity

- Two **words** are similar in meaning if their context vectors are similar

	aardvark	computer	data	pinch	result	sugar	…
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	

Materials

Stanford University

Pragmatics & Discourse

Pragmatics

Introduction

- Pragmatics studies (the origins), the uses and the effects of language.
- The study of the practical aspects of human action and thought.
- The study of the use of linguistic signs, words and sentences, in actual situations.
- It focuses on conversational implicature

Introduction

- The study of the practical aspects of human action and thought.
- The study of the use of linguistic signs, words and sentences, in actual situations.

Introduction

- Pragmatics is mainly the study of how the human brain understands natural language.
- People don’t say what they mean, or even normally.
- People typically use less words to express the scenario when some words are missing or implicit but humans usually understand the meaning in the context of the subject.
- Pragmatics is used to extract the information that is missing or implicit in natural language.

Introduction

- Speaker 1: there is a bookstore over there
- Speaker 2: no
- Speaker 1: why not
- Speaker 2: I‘m tired.

- Speaker 1: there is a bookstore over there. Let‘s go inside.
- Speaker 2: no, I don‘t want to go inside.
- Speaker 1: why do you not want to go inside?
- Speaker 2: I don‘t want to because I‘m tired.

Introduction

- Speaker 1:   there is a dog in the gate

Warning

…………………...

Beautiful

Speech Act Theory

- Austin founded speech act theory on the belief that speakers do not only utilize language to say things, but to do things.
- *When we use language to do something, we are performing a speech act.
- For example; Time out! Shotgun! Time Over! Start Now! etc….

Speech Act Theory

- Speech acts can also be performed with complete sentences

• John read the book.	*Assertion*
• Did John read the book?	*Question*
• Please pass the salt.	*Request*
• Kim's got a knife!	*Warning*
• Get out of here!	*Order*
• I will love you forever.	*Promise*
• I'll give you a reason to cry.	*Threat*

Discourse

Introduction

Discourse analysis study the ways sentences and utterances *(speech)* go together to make texts and interactions and how those texts and interactions fit into our social world.

It should be noticed also that discourse analysis is not just the study of language, but a way of looking at language as well.

Structural and functional definitions of discourse

- Structural or textual definition of discourse:

  Discourse is a particular unit of language (above the sentence).

- Functional definition of discourse: Discourse is a particular focus of language use.

Structural approach

- Find the constituents that have particular relationships with each other and that can occur in a restricted number of arrangements;
- Problems: units in which people speak do not always look like sentences, or grammatically correct sentences.

Example 1

Jack is tall and kind and don't hardly say anything. Love children. Respect his wife, Odessa, and all Odessa Amazon sisters (Celie’s Diary)

Functional approach to discourse

- Language performs six functions:
  - ✔ Addressor
  - ✔ Context
  - ✔ Addressee
  - ✔ Contact
  - ✔ Message
  - ✔ Code

(1) The Earth is round	---------------------------------	**Addressor**
(2) Tasty	---------------------------------	**Context**
(3) Get Out	---------------------------------	**Addressee**
(4) Hello!, Hi!"	---------------------------------	**Contact**
(5) What do you mean by	---------------------------------	**Message**
(6) Football	---------------------------------	**Code**

Spoken Discourse

In many ways, speech is not so different from writing:

1- When people speak they also produce different kinds of genres.
2- use different kinds of ‘social languages.
3- also promote particular versions of reality or ideologies

But there are some ways in which speech is very different from writing:

1- Speech is more interactive.
2- Speech tends to be more transient and spontaneous than writing.
3- While some genres like formal speeches and lectures are planned, most casual conversation is just made up as we go along.
4- Speech also usually takes place in some kind of physical context.

Spoken Discourse

1) Telephone conversations.
2) Television and cinema.
3) Instant messaging and text-based computer chats.