Beyond words: Morphology, Syntax, and Semantics

Northeastern University
CS 6120 Natural Language Processing

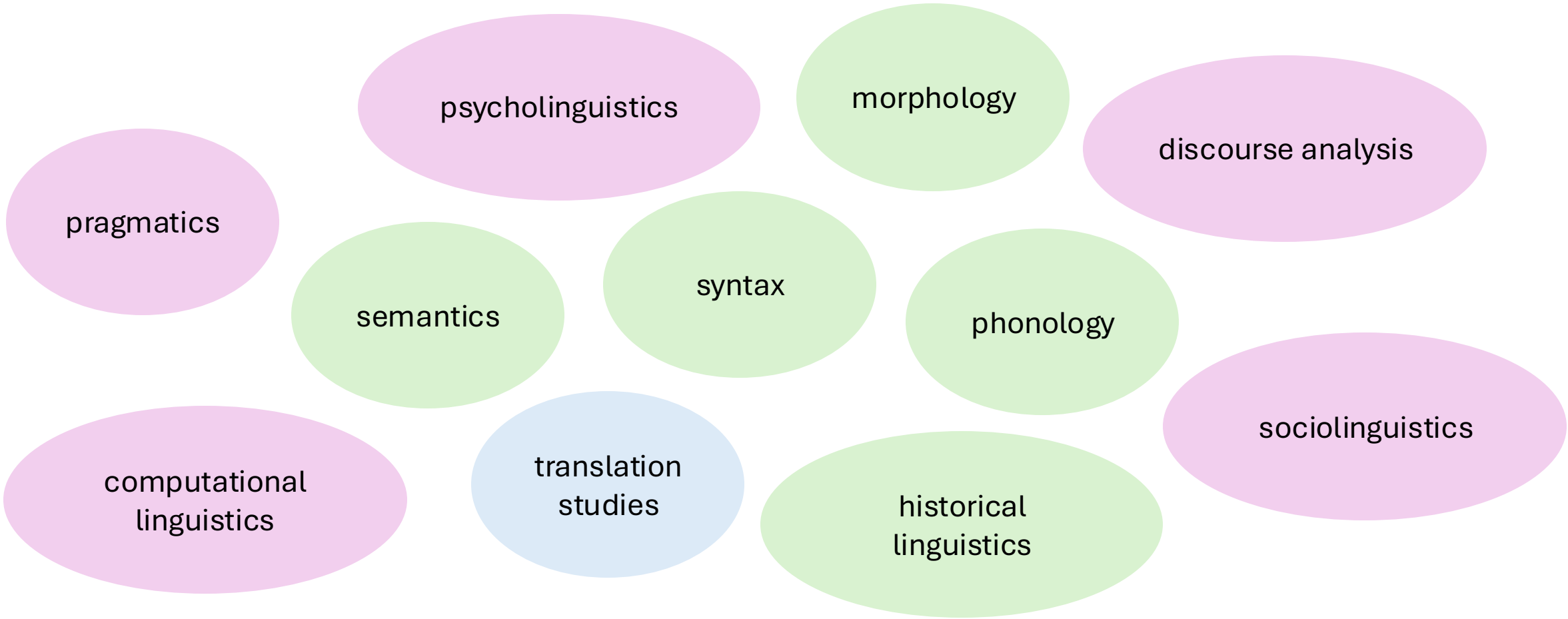
Si Wu

Some slides borrowed from Jurafsky & Martin Chapter 17

Logistics

- Coding assignment 2 is released. Due next Tuesday
  - Tips: submit on Gradescope early to prevent any issues with submission
- Instruction for project pitch is on the course website
- So far, we have covered and reviewed some foundational statistical models for NLP as well as neural network
- Today:
  - A taste of linguistics. Not just seeing words as probability and statistics.
  - We will talk about some interesting concepts in morphology, syntax, and semantics.
  - But also some classic computational methods for these linguistic tasks

Areas of linguistics



And so many more!

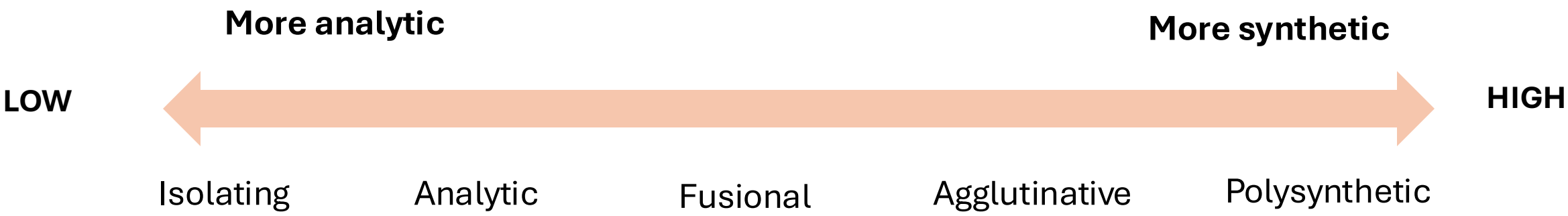
Morphology

How to form a word

Basic concepts

- Morphology: structure of words
- Morpheme: meaningful units in words that can't be divide further
  - Free morphemes: can stand on their own, e.g. run
  - Bound morphemes: must attach to other morphemes.
    - Affix
      - Prefix: go before. E.g. un-, over-
      - Suffix: go after. E.g. –ed, -s, -ing
      - Infix: go in between. E.g. the f word and bloody, abso-*f..ing*-lutely
      - Circumfix: both before or after, sandwiching a word. Not really in English , but maybe en-…-ed or un-…-able just to give an idea. Malay and Georgian has a lot of these.

Morpheme to word ratio



Chinese

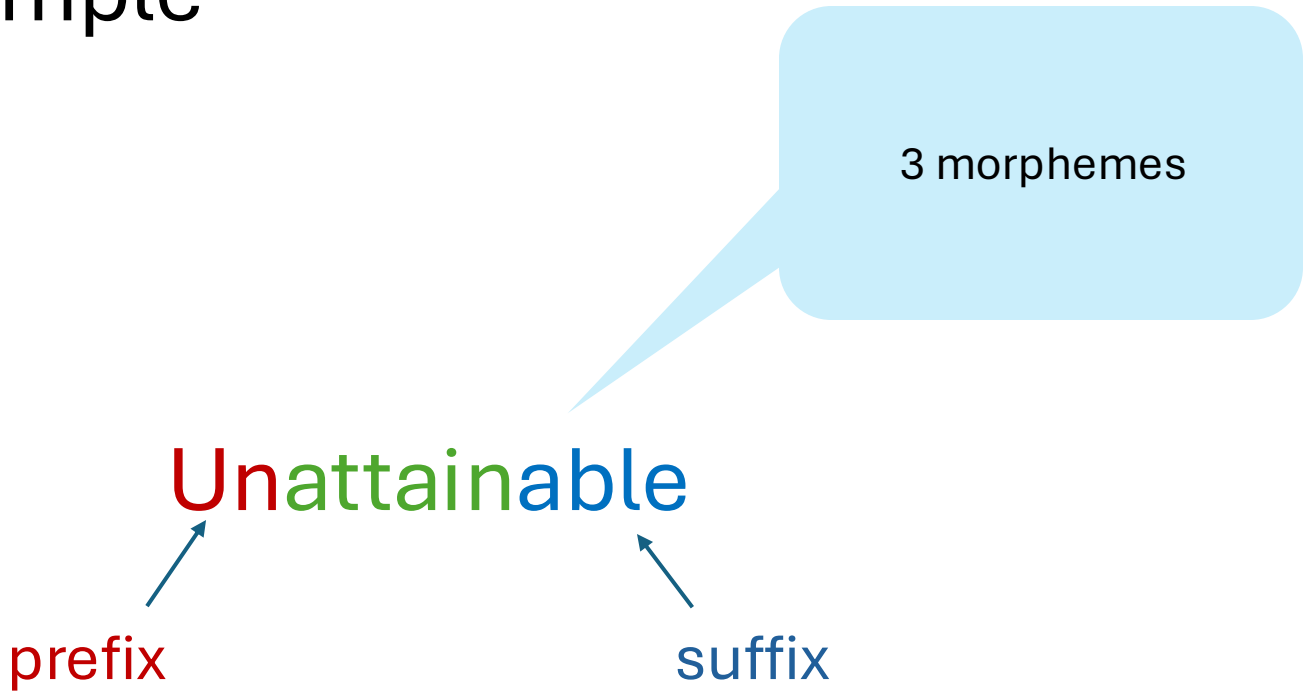
1 word 1 morpheme

English

Russian

A couple morphemes per word

English word example



German word example

3 morphemes

(or 4 if you further divide “Christmas” to “holy night”)

Weihnachtsbaumschmuck

Christmas tree Decoration, ornament jewelry

Chinese example

It’s a sentence not a word

2 morphemes,
1 word (词)

→ “it’s a small world!” what English speakers usually say

世界 （world） is a **calque** (literal word-for-word or root-for-root translation) created in Chinese Buddhism translations

original word is in Sanskrit लोकधातु (lokadhātu)
loka: world, realm → 世
dhatu: domain, element → 界

How is morphology useful in computational tasks?

- Lemmatization
  - Running, runs → run
- Information retrieval
  - Find different variations in query
    “a person running”, “a person who runs”, “a runner”
- Low-resource NLP
  - If we know the morphological pattern, we can generalize better

Lemmatize using WordNet’s built-in morphy function. Returns the input word unchanged if it cannot be found in WordNet.

```
>>> from nltk.stem import WordNetLemmatizer
>>> wnl = WordNetLemmatizer()
>>> print(wnl.lemmatize('dogs'))
dog
>>> print(wnl.lemmatize('churches'))
church
```

Syntax

How to form a sentence

What’s syntax?

- Syntax is the study of the structure of sentences and utterances
- Word order matters. Grammar matters too.
  - I love dogs, dogs loved by me, it was dogs that I love
  - ~~Water me drink~~

Constituent: a word or a group of words that function as a single unit.

- Often a phrase

Constituency testing

- **Constituency testing**: using grammatical tests and manipulations to determine whether something is a valid constituent
  - Replacement test
    - I write with a pen → I write with it
  - Movement test
    - I write with a pen → A pen that I write with
  - It-clefts
    - I write with a pen → It is a pen that I write with
  - Answer fragments, question test
    - I write with a pen → What do I write with? (a pen!)
  - Etc.

Syntax and different **sequence labeling** tasks

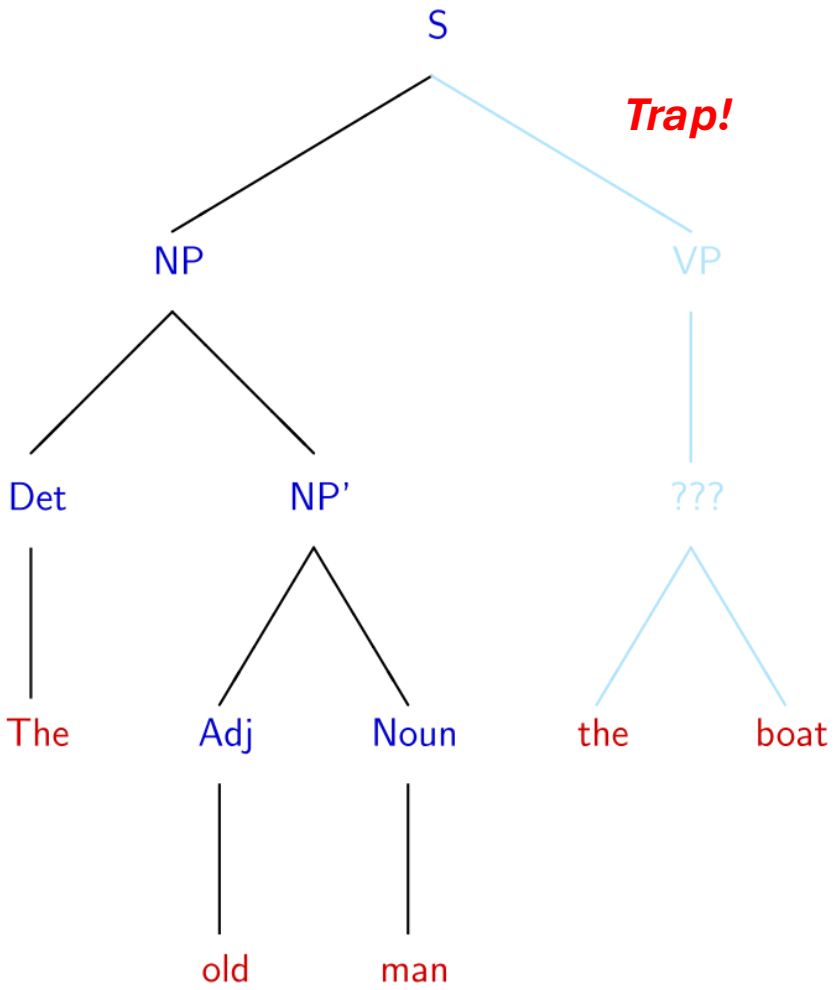
A variety of tasks in computational linguistics are related to studying and analyzing the structure of a sentence

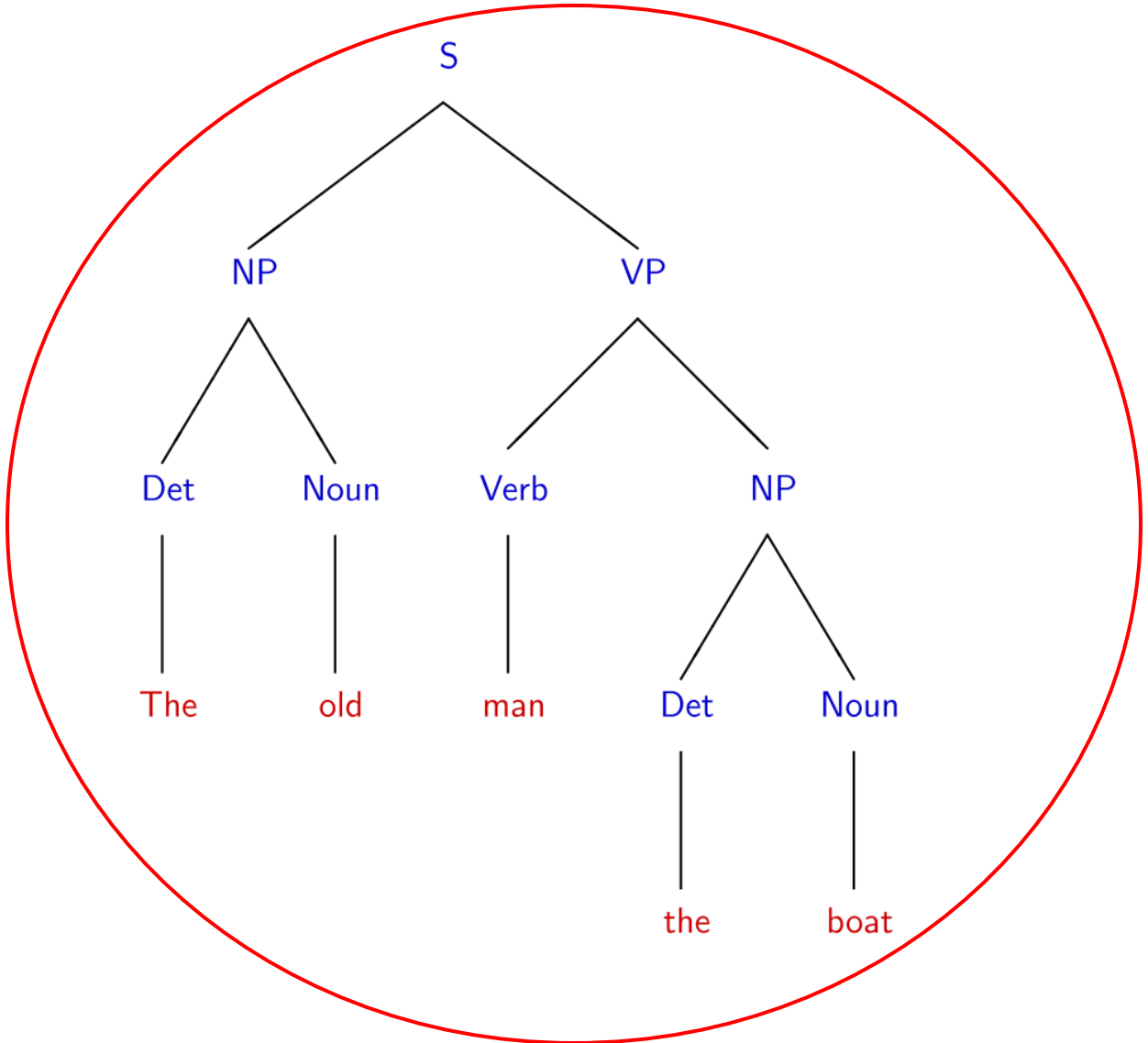
- **Part-of-speech tagging**: tag noun, verb, pronoun, adj, etc.
  - Knowing the tag of a word tells us what kind of tags will be its neighbors.
- **Named-entity recognition**: tag PERSON, LOCATION, ORGANIZATION
  - Knowing these proper nouns is important for question answering, stance detection, or information extraction.
- **Dependency parsing**: understand the relationship between words (head words and their dependents)
  - Help us understand meaning, which is helpful for many NLP tasks

Ambiguity of a sentence

- We’ve talked about ambiguity before. In written text, a word can have multiple meanings (polysemy), and a word can also have different PoS tags depending on the context.
- Moreover, a sentence can be read in different ways: garden path sentences
- This ambiguity makes many sequence labeling tasks difficult.

Garden path sentence: **The old man the boat**





I saw the man with the telescope

- Two legitimate interpretations→ ambiguous
  - I use the telescope to see the man
  - The man has a telescope, and I saw him

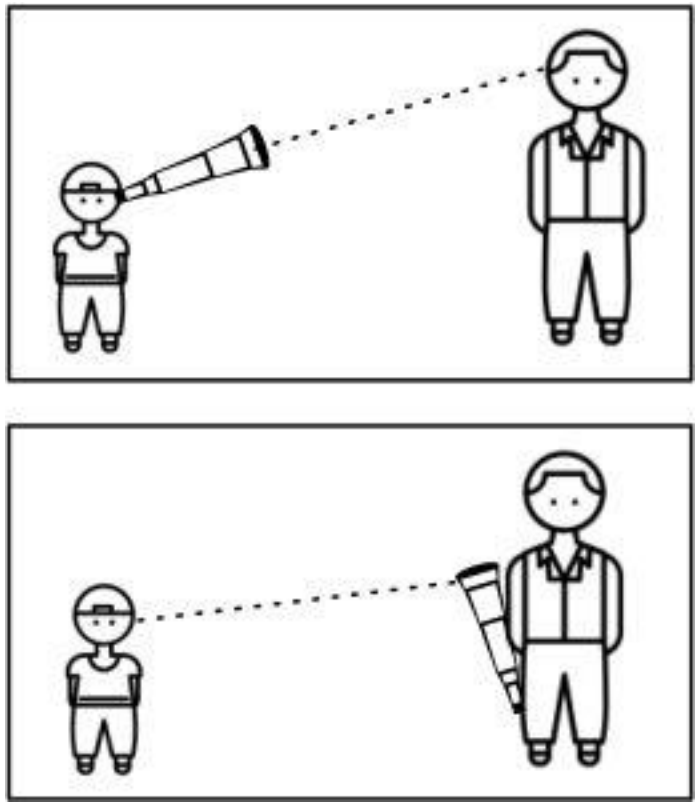


Figure from https://www.cs.jhu.edu/news/the-importance-of-ambiguity/

Part-of-speech tagging

Part of speech tagging is a little more fine-grained than lexical categories

→ grammatical categories

- Lexical categories
  - Noun,
  - verb,
  - adjective,
  - adverbs
    - temporal: now, then, yesterday;
    - locative: here, there;
    - manner: quickly
    - Linking: there, thus
    - Degree: very, too
    - Sentence: perhaps, honestly, technically
  - Preposition (postposition for other languages like Japanese and Turkish): under, around, behind
  - determiner (the, a, that, this, those),
  - pronounce (she, her),
  - conjunction (and, or, whenever),
  - numeral (one, twice),
  - interjection (ouch, tsk, damnit)

Knowing morphology we can…

- Create new words
  - Compounding existing words together: backpack, briefcase
  - Blending multiple words: staycation, mansplain
  - Clipping existing words: fam, cray
  - Changing PoS of an existing word: woke, I can’t even
  - Etc

A really cool museum in DC called Planet Word

Two classes of words: Open vs. Closed

- Closed class words
  - Relatively fixed membership
  - Usually **function** words: short, frequent words with grammatical function
    - determiners: ***a, an, the***
    - pronouns: ***she, he, I***
    - prepositions: ***on, under, over, near, by, …***
- Open class words
  - Usually **content** words: Nouns, Verbs, Adjectives, Adverbs
    - Plus interjections: **oh, ouch, uh-huh, yes, hello**
  - New nouns and verbs like *iPhone* or *to fax*

Open class ("content") words

Nouns

Proper

Janet
Italy

Common

cat, cats
mango

Verbs

Main

eat
went

Auxiliary

can
had

Adjectives *old green tasty*

Adverbs *slowly yesterday*

Numbers

122,312
one

Interjections *Ow hello*

… more

Closed class ("function")

Determiners *the some*

Conjunctions *and or*

Pronouns *they its*

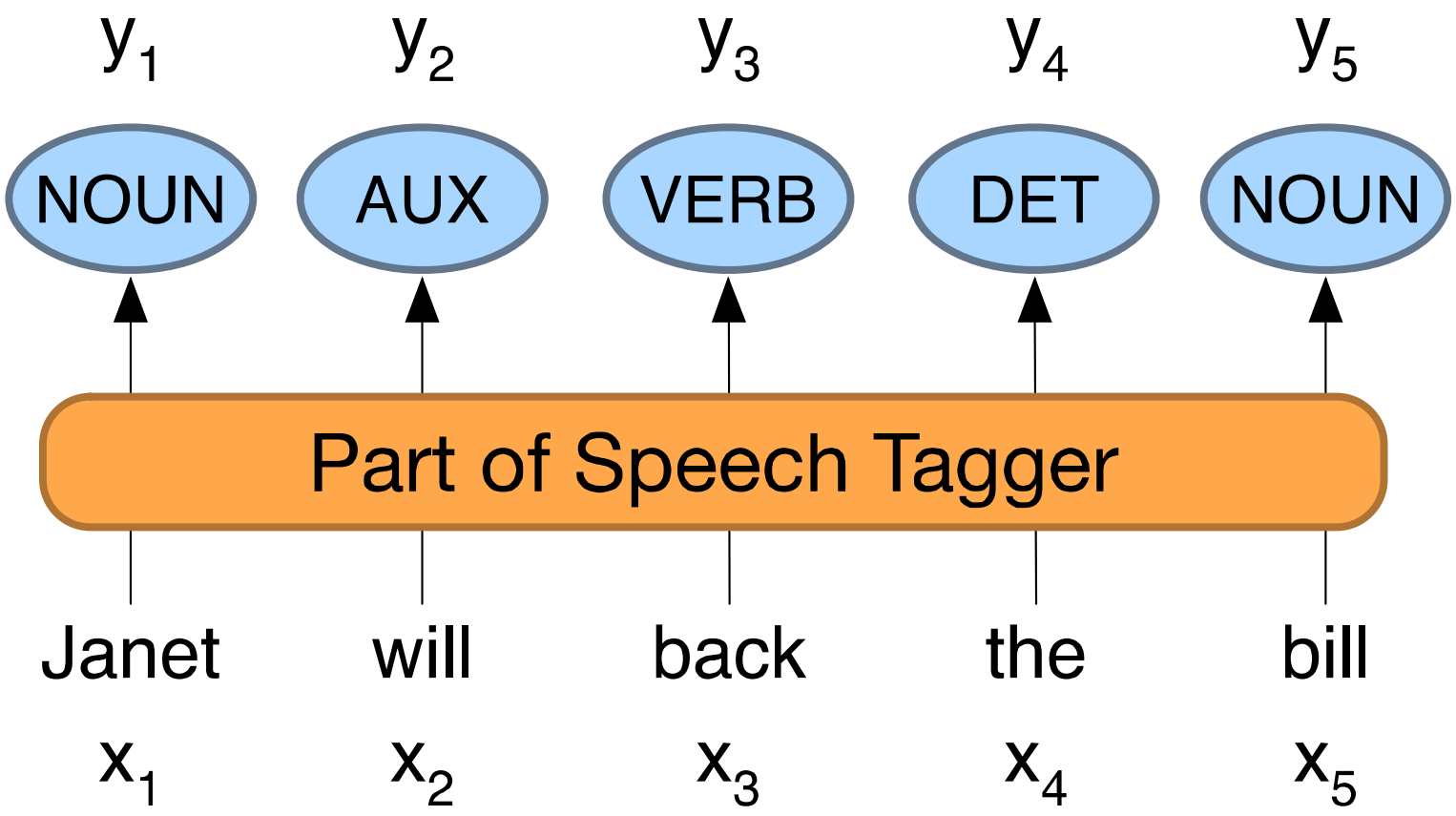
Prepositions *to with*

Particles *off up*

… more

Part-of-Speech Tagging

Map from sequence $x_1,\ldots,x_n$ of words to $y_1,\ldots,y_n$ of POS tags



Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coordinating conjunction	*and, but, or*	PDT	predeterminer	*all, both*	VBP	verb non-3sg present	*eat*
CD	cardinal number	*one, two*	POS	possessive ending	*'s*	VBZ	verb 3sg pres	*eats*
DT	determiner	*a, the*	PRP	personal pronoun	*I, you, he*	WDT	wh-determ.	*which, that*
EX	existential 'there'	*there*	PRP$	possess. pronoun	*your, one's*	WP	wh-pronoun	*what, who*
FW	foreign word	*mea culpa*	RB	adverb	*quickly*	WP$	wh-possess.	*whose*
IN	preposition/ subordin-conj	*of, in, by*	RBR	comparative adverb	*faster*	WRB	wh-adverb	*how, where*
JJ	adjective	*yellow*	RBS	superlatv. adverb	*fastest*	$	dollar sign	*$*
JJR	comparative adj	*bigger*	RP	particle	*up, off*	#	pound sign	*#*
JJS	superlative adj	*wildest*	SYM	symbol	+,%, &	"	left quote	' or "
LS	list item marker	*1, 2, One*	TO	"to"	*to*	"	right quote	' or "
MD	modal	*can, should*	UH	interjection	*ah, oops*	(	left paren	[, (, {, <
NN	sing or mass noun	*llama*	VB	verb base form	*eat*	)	right paren	],), }, >
NNS	noun, plural	*llamas*	VBD	verb past tense	*ate*	,	comma	,
NNP	proper noun, sing.	*IBM*	VBG	verb gerund	*eating*	.	sent-end punc	. ! ?
NNPS	proper noun, plu.	*Carolinas*	VBN	verb past part.	*eaten*	:	sent-mid punc	: ; ... – -

Figure 8.1 Penn Treebank part-of-speech tags (including punctuation).

Working through an example

Nicole was born in Hawaii

NNP	VBD	VBN	IN	NNP
(proper noun)	(verb, past tense)	(verb, past participle)	(preposition)	(proper noun)

Why Part of Speech Tagging?

- Can be useful for other NLP tasks
  - Parsing: POS tagging can improve syntactic parsing
  - MT: reordering of adjectives and nouns (say from Spanish to English)
  - Sentiment or affective tasks: may want to distinguish adjectives or other POS
  - Text-to-speech (how do we pronounce “lead” or "object"?)
- Or linguistic or language-analytic computational tasks
  - Need to control for POS when studying linguistic change like creation of new words, or meaning shift
  - Or control for POS in measuring meaning similarity or difference

How difficult is POS tagging in English?

Roughly 15% of word types are ambiguous

- Hence 85% of word types are unambiguous
- *Janet* is always proper noun, *hesitantly* is always adverb

But those 15% tend to be very common.

So ~60% of word tokens are ambiguous

E.g., *back*

earnings growth took a back/ADJ seat
a small building in the back/NOUN
a clear majority of senators back/VERB the bill
enable the country to buy back/PART debt
I was twenty-one back/ADV then

POS tagging performance in English

- How many tags are correct? (Tag accuracy)
  - About 97%
    - Hasn't changed in the last 10+ years
    - HMMs, CRFs, BERT perform similarly .
    - Human accuracy about the same
- But baseline is 92%!
  - Baseline is performance of stupidest possible method
    - "Most frequent class baseline" is an important baseline for many tasks
      - Tag every word with its most frequent tag
      - (and tag unknown words as nouns)
  - Partly easy because
    - Many words are unambiguous

Sources of information for POS tagging

Janet will back the bill

AUX/NOUN/VERB? NOUN/VERB?

- Prior probabilities of word/tag
  - "will" is usually an auxiliary that help verb make sense
- Identity of neighboring words
  - "the" means the next word is probably not a verb
- Morphology and wordshape:
  - Prefixes unable: un- → adj
  - Suffixes importantly: -ly → adj
  - Capitalization Janet: CAP → proper noun

Standard algorithms for POS tagging

Supervised Machine Learning Algorithms:

- Hidden Markov Models
- Conditional Random Fields (CRF)/ Maximum Entropy Markov Models (MEMM)
- Neural sequence models (RNNs or Transformers)
- Large Language Models (like BERT), finetuned

All required a hand-labeled training set, all about equal performance (97% on English)

All make use of information sources we discussed

- Via human created features: HMMs and CRFs
- Via representation learning: Neural LMs

Hidden Markov and PoS

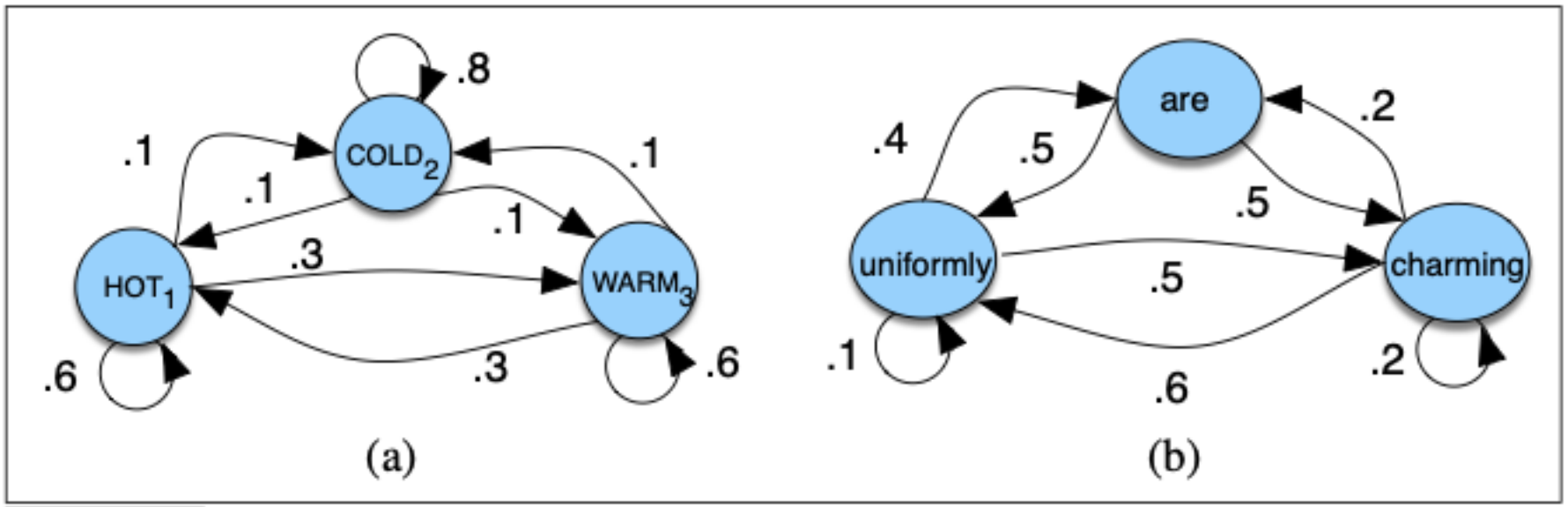


Figure A.1 A Markov chain for weather (a) and one for words (b), showing states and transitions. A start distribution π is required; setting $\pi = [0.1, 0.7, 0.2]$ for (a) would mean a probability 0.7 of starting in state 2 (cold), probability 0.1 of starting in state 1 (hot), etc.

- Observed event: words in the input
- Hidden even: PoS tags

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t (drawn from a vocabulary $V = v_1, v_2, ..., v_V$) being generated from a state q_i
$\pi = \pi_1, \pi_2, ..., \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

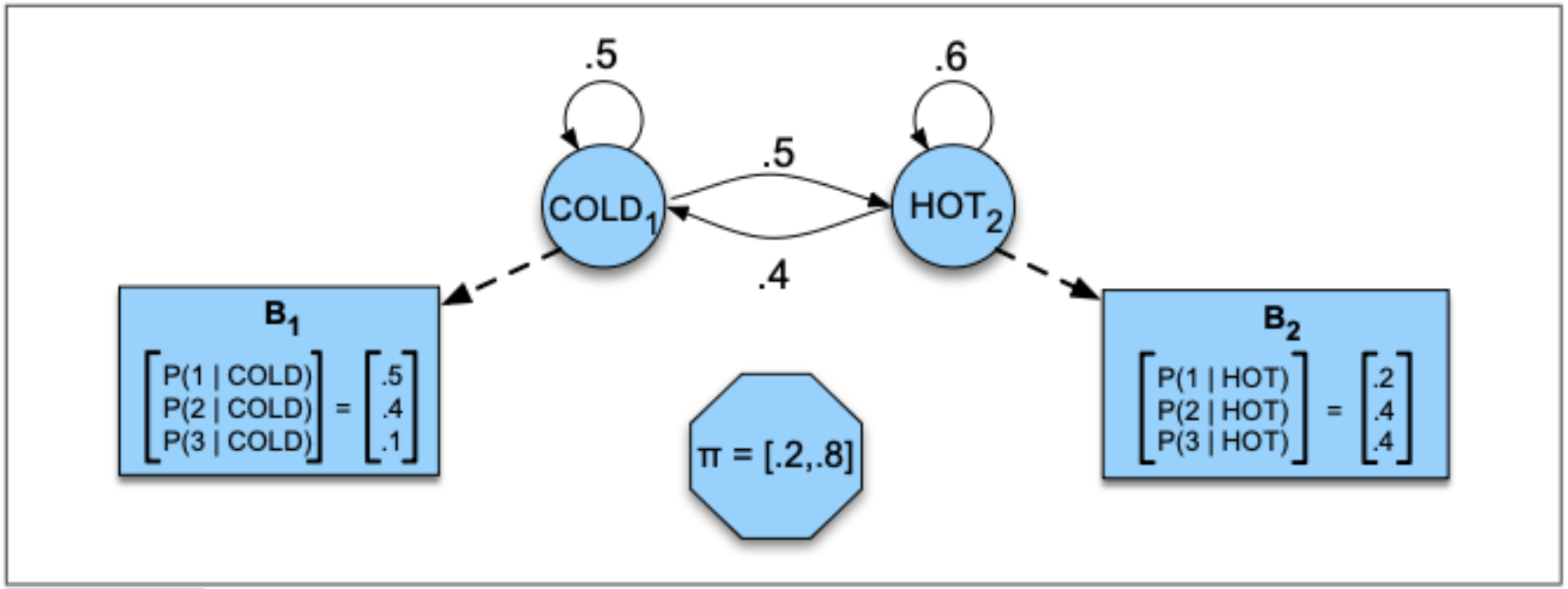


Figure A.2 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

Viterbi algorithm

- For decoding HMM → **find the most likely tag** sequence for a sentence
- A dynamic programming algorithm
  - Like Dijkstra, it computes the shortest path

function VITERBI(*observations* of len T,*state-graph* of len N) **returns** *best-path*, *path-prob*

create a path probability matrix *viterbi[N,T]*
for each state s **from** 1 **to** N **do** ; initialization step
    $viterbi[s,1] \leftarrow \pi_s * b_s(o_1)$
    $backpointer[s,1] \leftarrow 0$
for each time step t **from** 2 **to** T **do** ; recursion step
  for each state s **from** 1 **to** N **do**
    $viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
    $backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
$bestpathprob \leftarrow \max_{s=1}^{N} viterbi[s,T]$; termination step
$bestpathpointer \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T]$; termination step
bestpath $\leftarrow$ the path starting at state *bestpathpointer*, that follows backpointer[] to states back in time
return *bestpath*, *bestpathprob*

Baum-Welch or forward-backward algorithm

- For training with HMM
- A special case of expectation-maximization (EM) algorithm.
- It will train both the **transition** probabilities A and the **emission** probabilities B of the HMM

function FORWARD-BACKWARD(*observations* of len T, *output vocabulary* V, *hidden state set* Q) **returns** *HMM=(A,B)*

initialize A and B

iterate until convergence

E-step

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\alpha_T(q_F)} \quad \forall\, t \text{ and } j$$

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij} b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(q_F)} \quad \forall\, t,\ i, \text{ and } j$$

M-step

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1}\sum_{k=1}^{N} \xi_t(i,k)}$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 s.t.\ O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

return A, B

Named-entity recognition

- What is **named entity**?
  - In its core usage, means anything that can be referred to with a proper name.
  - Most common 4 tags:
    - PER (Person): “Marie Curie”
    - LOC (Location): “New York City”
    - ORG (Organization): “Stanford University”
    - GPE (Geo-Political Entity): "Boulder, Colorado”

- Named entities are often multi-word phrases
  - But the term is also extended to things that aren't entities:
    - dates, times, prices

Named Entity tagging

The task of named entity recognition (NER):

- find spans of text that constitute proper names
- tag the type of the entity.

Type	Tag	Sample Categories	Example sentences
People	PER	people, characters	**Turing** is a giant of computer science.
Organization	ORG	companies, sports teams	The **IPCC** warned about the cyclone.
Location	LOC	regions, mountains, seas	**Mt. Sanitas** is in **Sunshine Canyon**.
Geo-Political Entity	GPE	countries, states	**Palo Alto** is raising the fees for parking.

Figure 17.5 A list of generic named entity types with the kinds of entities they refer to.

NER output

Citing high fuel prices, [$_{\text{ORG}}$ **United Airlines**] said [$_{\text{TIME}}$ **Friday**] it has increased fares by [$_{\text{MONEY}}$ **$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{\text{ORG}}$ **American Airlines**], a unit of [$_{\text{ORG}}$ **AMR Corp.**], immediately matched the move, spokesman [$_{\text{PER}}$ **Tim Wagner**] said. [$_{\text{ORG}}$ **United**], a unit of [$_{\text{ORG}}$ **UAL Corp.**], said the increase took effect [$_{\text{TIME}}$ **Thursday**] and applies to most routes where it competes against discount carriers, such as [$_{\text{LOC}}$ **Chicago**] to [$_{\text{LOC}}$ **Dallas**] and [$_{\text{LOC}}$ **Denver**] to [$_{\text{LOC}}$ **San Francisco**].

Working through an example

Nicole was born in Hawaii

PER LOC

Type	Tag	Sample Categories	Example sentences
People	PER	people, characters	**Turing** is a giant of computer science.
Organization	ORG	companies, sports teams	The **IPCC** warned about the cyclone.
Location	LOC	regions, mountains, seas	**Mt. Sanitas** is in **Sunshine Canyon**.
Geo-Political Entity	GPE	countries, states	**Palo Alto** is raising the fees for parking.

Figure 17.5 A list of generic named entity types with the kinds of entities they refer to.

Why NER?

- Sentiment analysis: consumer's sentiment toward a particular company or person?
- Question Answering: answer questions about an entity?
- Information Extraction: Extracting facts about entities from text.

Why NER is hard

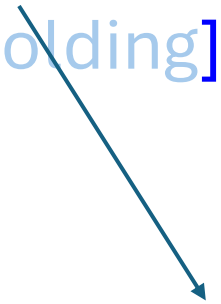
1) Segmentation
   - In POS tagging, no segmentation problem since each word gets one tag.
   - In NER we have to find and segment the entities!
2) Type ambiguity

[PER Washington] was born into slavery on the farm of James Burroughs.
[ORG Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [LOC Washington] for what may well be his last state visit.
In June, [GPE Washington] passed a primary seatbelt law.

BIO Tagging

- How can we turn this structured problem into a sequence problem like POS tagging, with one label per word?

[PER Jane Villanueva] of [ORG United] , a unit of [ORG United Airlines Holding] , said the fare applies to the [LOC Chicago] route.



[**Begin** a span] Jane
[**Inside** a span] Villanueva

BIO Tagging

B: token that *begins* a span

I: tokens *inside* a span

O: tokens outside of any span

So if we have 2 types of **entity**:

PER and ORG. → n = 2,

n is # of types of entities

Then since each entity type (e.g. PER) have have B and/or I tags, and O tag outside of the span

In total, we have ***2n+1*** *tags*

B-PER, I-PER
B-ORG, I-ORG
O

Words	BIO Label
Jane	B-PER
Villanueva	I-PER
of	O
United	B-ORG
Airlines	I-ORG
Holding	I-ORG
discussed	O
the	O
Chicago	B-LOC
route	O
.	O

BIO Tagging variants: IO and BIOES

- [PER Jane Villanueva] of [ORG United] , a unit of [ORG United Airlines Holding] , said the fare applies to the [LOC Chicago] route.

Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O

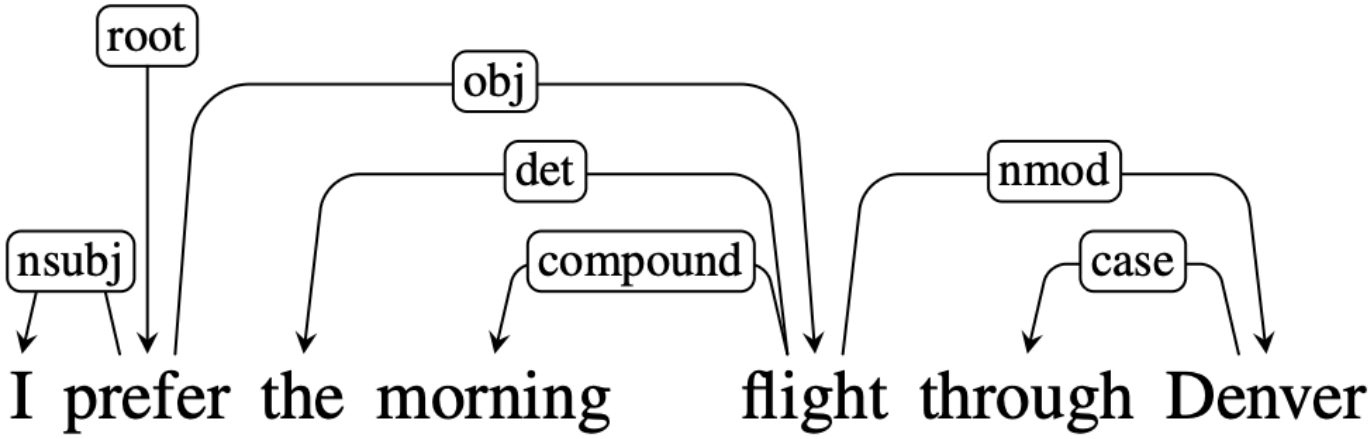
Standard algorithms for NER

Supervised Machine Learning given a human-labeled training set of text annotated with tags

- Hidden Markov Models
- Conditional Random Fields (CRF)/ Maximum Entropy Markov Models (MEMM)
- Neural sequence models (RNNs or Transformers)
- Large Language Models (like BERT), finetuned

Dependency Parsing

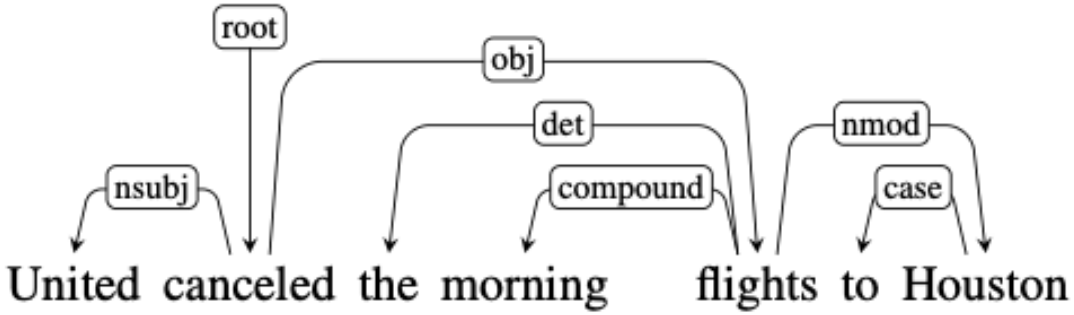
- Dependency grammar: sentences are structured around words and their dependencies.
- Relations among the words are illustrated with directed, labeled arcs from **heads** to **dependents**. We call this a **typed dependency structure**.
  - "typed" because the labels are from a fixed set of grammatical relations



Dependency relations

- Different classes of grammatical relations/function between head and dependent
  - Subject, direct object, indirect object

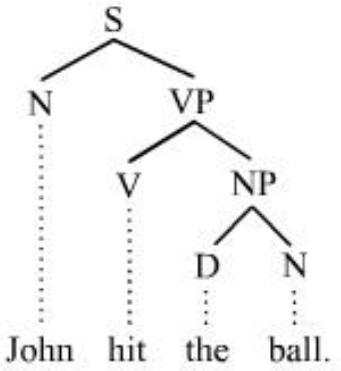
- **Universal dependencies** (UD) project (de Marneffe et al., 2021) is an open community effort to annotate dependencies and other aspects of grammar across more than 100 languages, provides an inventory of 37 dependency relations.



Clausal Argument Relations	Description
NSUBJ	Nominal subject
OBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Figure 19.2 Some of the Universal Dependency relations (de Marneffe et al., 2021).

Dependency tree



Constituency-based parse tree

- A directed graph illustrates a dependency grammar analysis
- Nodes → words
- Edges → labeled dependency relations (nsubj, aux, obl)
- It's a tree!
  - Each word only has one parent
  - The tree itself encodes syntactic relations described by dependency grammar

- Different from Phrase Structure Grammar
  - PSG groups words into phrases, but Dependency Grammar is between individual words
    - But related: a constituent / phrase in PSG is probably a subtree in dependency tree

Semantics

How the meaning is formed

What is semantics?

- The study of meaning
- Different from **pragmatics**, which is the meaning in a given (conversational) context

- Ambiguity again!
  - Lexical ambiguity: what word sense am I using?
  - Structural syntactic ambiguity: how to parse the sentence?
  - Semantic ambiguity: different interpretations of a sentence
    - “I saw the man with a telescope”

Entailment

- If X is true, then Y is definitely true”
  - If Nicole was born in Hawaii, then Nicole was born
- X entails Y doesn’t *mean* Y entails X
- Entailment do not need conversational context

John won the game → John played the game

Pragmatic: Cooperative principle

- In daily life, we communicate with goals in mind, and these cooperative principle help us achieve these goals

- **Gricean's maxims**:
  - Maxim of quantity : just enough information but not too much
  - Maxim of quality: tell the truth, don't say what you don't know
  - Maxim of relation: your information should be relevant
  - Maxim of manner: be clear about your information, avoid ambiguity, avoid obscurity, don't be unncessarily wordy, give information in a order that makes sense
- Flouting the maxims → could be sarcasm or irony!
- Violating the maxims (lying, misleading, etc.) …. Hmmm… If intentionally… just a bad person

Pointers

- Hidden Markov Model: notes by Michael Collin https://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/hmms.pdf
- Other interesting syntactic tasks are dependency parsing, semantic role labeling. You can read more about it in Jurafsky and Martin Chapter 19 and Chapter 21.
- If you are interested, we can cover interesting things on pragmatic (how context contributes to meaning) and speech act. (More linguistics yeah!)

- Next lecture we will talk about RNN and LSTM