PennState College of the Liberal Arts

C-SoDA Center for Social Data Analytics

Day 2 - NLP pipelines and core NLP tasks

Advanced Text as Data: Natural Language Processing
Essex Summer School in Social Science Data Analysis

Burt L. Monroe (Instructor) & Sam Bestvater (TA)
Pennsylvania State University

July 27, 2021

Today

- NLP “annotation” pipelines (core “processing” tasks for which there are multiple decent solutions)
  - Tokenization / segmentation
  - Normalization / lemmatization / stemming / morphology
  - Sequence labeling — parts of speech (POS), named entity recognition (NER)
  - Dependency parsing
- Demo: NLP pipelines in R and Python

Tokenization and Segmentation

Tokenization

- Text is just a sequence of characters (bytes). How do we split it into words and sentences?
- What’s a word / word boundaries.
- Sentence boundaries.

White space and punctuation … what’s the problem?

- m.p.h., Ph.D., AT&T, D.C., Mrs.
- R2-D2, SARS-Cov-2, New York-based
- $12.52, 07/27/21, @burtmonroe, #blessed, !!!
- we’re, couldn’t’ve, l’honneur, j’ai
- New York, Supreme Court, web site, website
- Vehkehrswegeplanungsbeschleunigungsgesetzen (laws for the acceleration of traffic route planning)
- uygarlaştıramadıklarımızdanmışsınızcasına
  uygar_laş_tır_ama_dık_lar_ımız_dan_mış_sınız_casına (as if you are among those we were not able to cause to be civilized)
- *Chinese:* 我开始写小说 = 我 开始 写 小说
  I start(ed) writing novel(s)

姚明进入总决赛 “Yao Ming reaches the finals”

3 words?

姚明	进入	总决赛
YaoMing	reaches	finals

5 words?

姚	明	进入	总	决赛
Yao	Ming	reaches	overall	finals

7 characters? (don't use words at all):

姚	明	进	入	总	决	赛
Yao	Ming	enter	enter	overall	decision	game

"The San Francisco-based restaurant," they said, "doesn’t charge $10".

Editable Code spaCy v3.0 · Python 3 · via Binder

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp('"The San Francisco-based restauran
for token in doc:
    print(token.text)
```

RUN

```
"
The
San
Francisco
-
based
restaurant
,
"
they
said
,
"
does
n't
charge
$
10
"
.
```

spaCy default

```
Francisco-based
Francisco - based

"   doesn't
"   doesn'   t
"doesn't
"   does     n't

$ 10 " .
$ 10 "
$10"
```

“_ The _ San _ Francisco-based _ restaurant _ , _ ” _
they_ said_ ,_ "_ does _ n’t _ charge_ $_ 10 _ " _ . _

Penn Treebank 3 standard

TreebankWordTokenizer

1. `` | The | San | Francisco-based | restaurant | , | '' | they | said | , | `` | doesn't | charge | $ | 10 | '' | .

WordPunctTokenizer

1. " | The | San | Francisco | - | based | restaurant | ," | they | said | , | " | doesn | ' | t | charge | $ | 10 | ".

PunktWordTokenizer

1. " | The | San | Francisco-based | restaurant | , | " | they | said | , | " | doesn't | charge | $10 | " | .

WhitespaceTokenizer

1. "The | San | Francisco-based | restaurant," | they | said, | "doesn't | charge | $10".

pattern

1. " | The | San | Francisco-based | restaurant | , | " | they | said | , | " | doesn't | charge | $ | 10 | " | .

nltk options

Sentence Tokenization/Segmentation

- For the most part, bag-of-words methods don't care at all about the "sentence." What matters is "what's a term" and "what's a document?" (the latter being an unappreciated question).

- For the most part, traditional NLP doesn't care about anything else.

- But recognizing or defining sentences isn't trivial, either.

Dr. Jane R. Smith, Ph.D., lives 3.5 miles from D.C. Mr. J. E. Jones lives in the U.K.

"The San Francisco-based restaurant," they said, "doesn't charge $10".

Very small crowds, you know it, they know it, we all know it. ("One" sentence?)
Highly respected man. Four-star general. ("Two" sentences?)

Can you have a legitimate sentence without a verb? What? Yes!

Tokens, Types, and Vocabulary

- Important difference between tokens and types.
- Types are the unique tokens — they constitute the vocabulary, V.
- Zipf’s law, etc., … we have many rare tokens and great sparsity.
- Out-of-vocabulary (OOV) problem
    - <UNK> token
    - The hashing trick
    - Subword tokenization

The hashing trick

- Choose some method for mapping any token (any sequence of bytes) to an integer, like adding the byte values of their characters.
- Map that integer into an integer in a fixed range using modulo arithmetic (like a clock).
- Use those integers as features.
- Now every possible token maps to an existing feature/input.
    - Collisions. Degrade performance and complicate interpretation.

Many of the state-of-the-art use subword tokenization

- BERT uses WordPiece tokenization
- RoBERTa, GPT-2, XLM use Byte Pair Encoding variants

Are these morphemes (smallest meaning-bearing units) as often claimed? Does it matter?

Search this file…

bert-base-cased	bert-base-uncased	bert-base-multilingual-cased	gpt2	xlm-mlm-en-2048
Marion	marion	Marion	Mar-ion	marion
b-ap-tist	baptist	ba-ptis-t	b-apt-ist	baptist
n-ug-gets	nu-gg-ets	nu-gge-ts	n-uggets	nu-g-gets

tokenisations.csv hosted with ❤️ by GitHub view raw

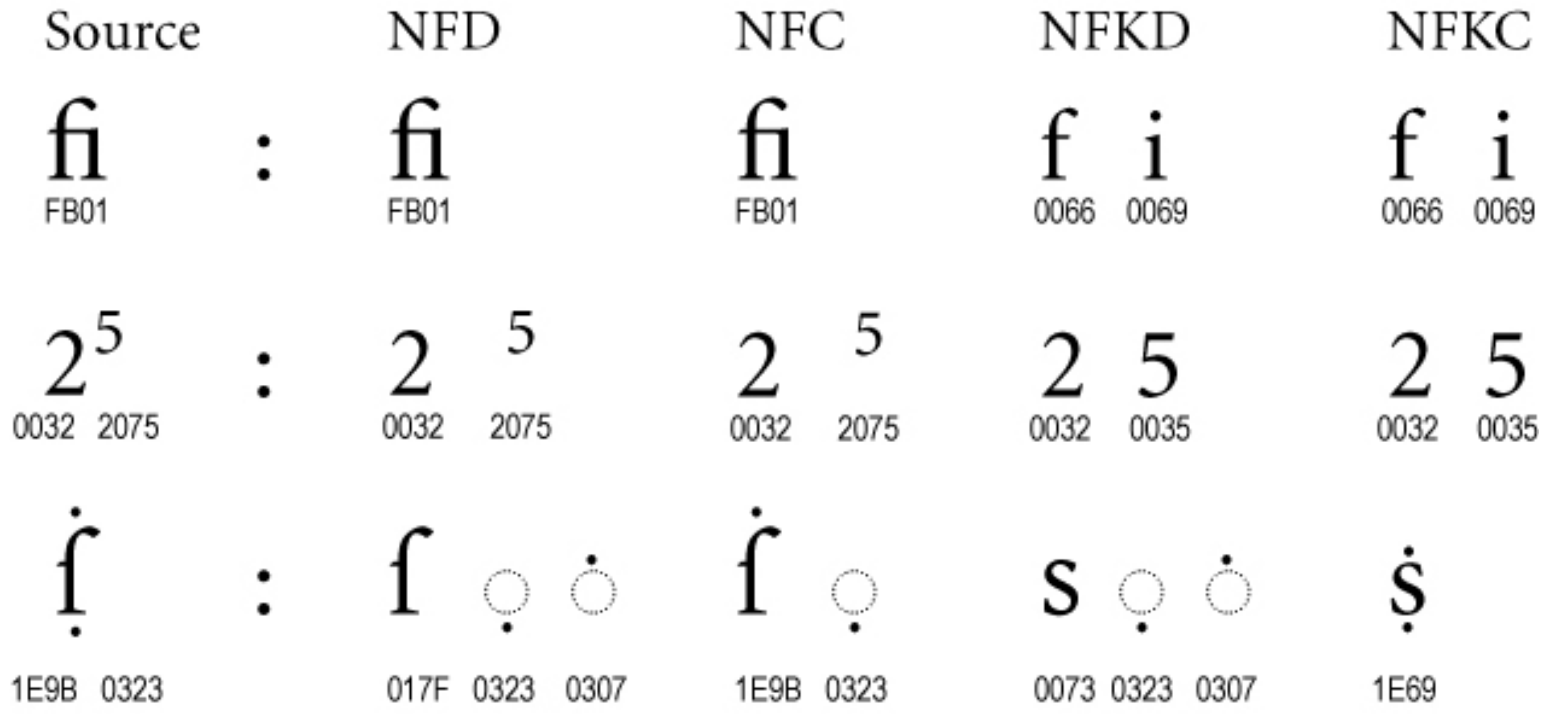
The different tokenization of the words “Marion”, “baptist” and “nuggets”

Source: Gergely Nemuth. 2019. “Comparing Transformer Tokenizers.”

Normalization

Normalization of character sets

- Limited character sets, e.g. ASCII? (Pairs well with “exact match” voter laws for disenfranchising voters with accents in their names!)
- Unicode normalization



Normalization - “Pre-processing”

- Case-folding (“lower casing”)
  - Good for search engines
  - Good for topic models?
  - Bad for named entity recognition / information extraction?
  - Do it **after** sentence segmentation!
- Spelling correction?

Normalization - Morphology

- A wordform is a word fully inflected as it appears in running text
- A lemma is an uninflected root of any given wordform. (so: “A wordform be a word full inflect as it appear run text.”)
- Lemmatization — tagging a token with its lemma
- Involves morphological parsing. Wordforms consist of morphemes (meaningful subword units)
  - stems - core meaning-bearing units - generally “free morphemes”
  - affixes - prefixes/suffixes, often with grammatical functions. “bound morphemes”
- Stemming: Crude algorithmic approximation

The Porter stemmer at work

This was not the map we found in Billy Bones's chest, but an accurate copy, complete in all things-names and heights and soundings-with the single exception of the red crosses and the written notes.

→

Thi wa not the map we found in Billi Bone s chest but an accur copi complet in all thing name and height and sound with the singl except of the red cross and the written note .

Source: Jurafsky & Martin, SLP3 slides

How many different words are there?

Inflection creates different forms of the same word:

Verbs: to be, being, I am, you are, he is, I was,
Nouns: one book, two books

Derivation creates different words from the same lemma:

grace ⇒ disgrace ⇒ disgraceful ⇒ disgracefully

Compounding combines two words into a new word:

cream ⇒ ice cream ⇒ ice cream cone ⇒ ice cream cone bakery

Word formation is productive:

New words are subject to all of these processes:
Google ⇒ Googler, to google, to ungoogle, to misgoogle, googlification, ungooglification, googlified, Google Maps, Google Maps service,...

Source: Julia Hockenmaier, Illinois CS447 slides

Dealing with complex morphology is necessary for many languages

- e.g., the Turkish word:
- Uygarlastiramadiklarimizdanmissinizcasina
- `(behaving) as if you are among those whom we could not civilize’
- Uygar `civilized’ + las `become’

+ tir `cause’ + ama `not able’

+ dik `past’ + lar ‘plural’

+ imiz ‘p1pl’ + dan ‘abl’

+ mis ‘past’ + siniz ‘2pl’ + casina ‘as if’

Source: Jurafsky & Martin, SLP3 slides (K. Oflazer p.c.)

N-gram language models

1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, ’tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv’d in; –It cannot be but so.

N = 884,647 tokens, |V| = 29,066

300,000 bigrams observed out of 844 million possible: 99.96% zeros

What about 4-grams?

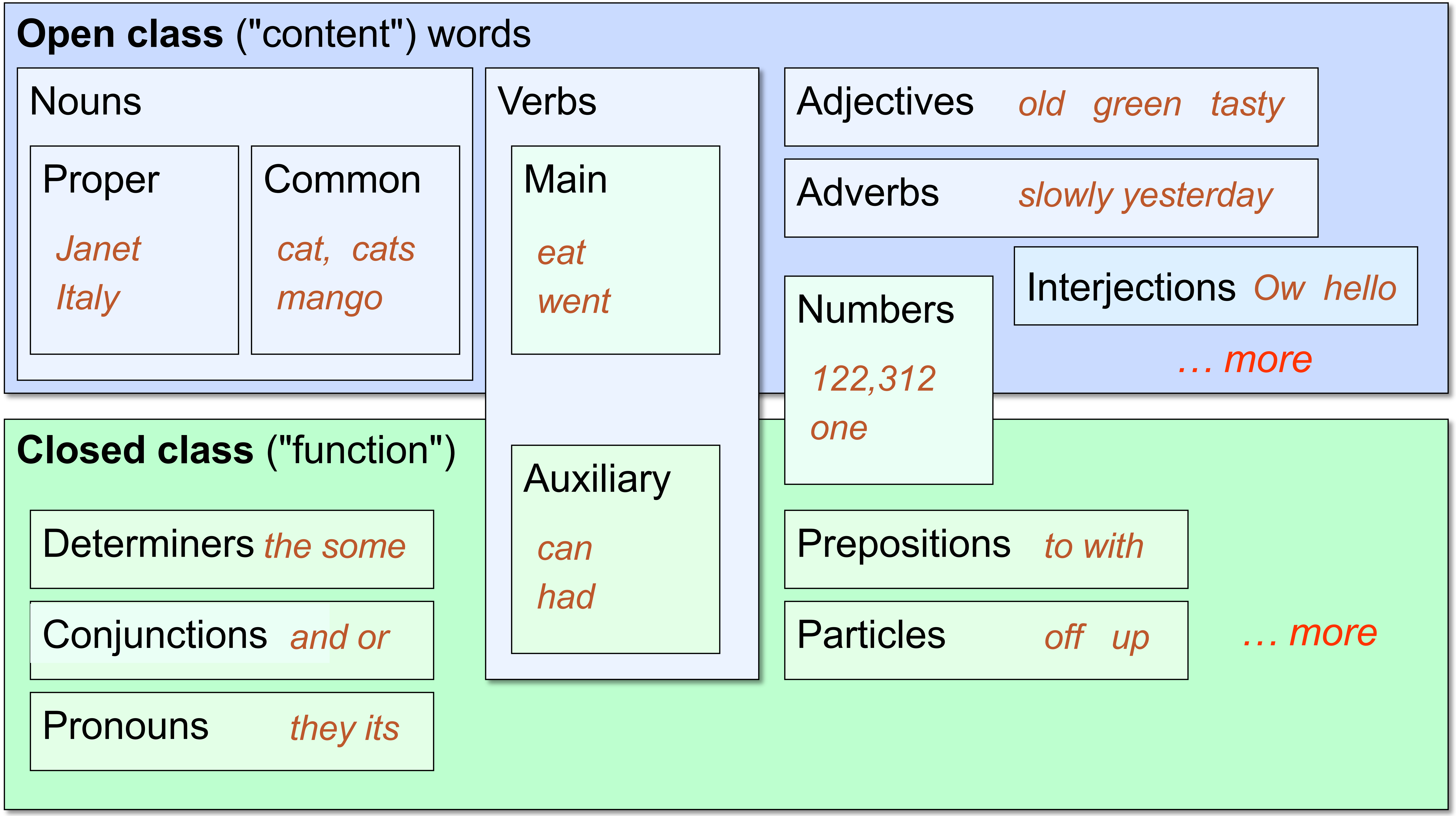
It looks like Shakespeare because it is! Overfitting!!!

Source: Jurafsky & Martin, SLP3

Zeros are a problem

- Generalization - training data doesn't look like the test set
- Zeros in the training data *can't* predict nonzeros in the test set.
- Smoothing = Bayesian prior = regularization = "add a little bit to the zeros"
- pseudo-counts / "hallucinated counts"
- Simplistic approach: Laplace smoothing — add 1 to everything.

Part-of-Speech Tagging



Source: Jurafsky & Martin, SLP3 slides

Universal POS tags

from Universal Dependencies (Nivre et al 2016)

	Tag	Description	Example
Open Class	**ADJ**	Adjective: noun modifiers describing properties	*red*, *young*, *awesome*
	ADV	Adverb: verb modifiers of time, place, manner	*very*, *slowly*, *home*, *yesterday*
	NOUN	words for persons, places, things, etc.	*algorithm*, *cat*, *mango*, *beauty*
	VERB	words for actions and processes	*draw*, *provide*, *go*
	PROPN	Proper noun: name of a person, organization, place, etc..	*Regina*, *IBM*, *Colorado*
	INTJ	Interjection: exclamation, greeting, yes/no response, etc.	*oh*, *um*, *yes*, *hello*
Closed Class Words	**ADP**	Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation	*in, on, by under*
	AUX	Auxiliary: helping verb marking tense, aspect, mood, etc.,	*can, may, should, are*
	CCONJ	Coordinating Conjunction: joins two phrases/clauses	*and*, *or*, *but*
	DET	Determiner: marks noun phrase properties	*a, an, the, this*
	NUM	Numeral	*one, two, first, second*
	PART	Particle: a preposition-like form used together with a verb	*up, down, on, off, in, out, at, by*
	PRON	Pronoun: a shorthand for referring to an entity or event	*she, who, I, others*
	SCONJ	Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement	*that*, *which*
Other	**PUNCT**	Punctuation	; , ()
	SYM	Symbols like $ or emoji	$, %
	X	Other	asdf, qwfg

Table source: Jurafsky and Martin, *SLP3*, 2021.

Penn Treebank POS tags

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

Figure 8.2 Penn Treebank part-of-speech tags.

Table source: Jurafsky and Martin, *SLP3*, 2021.

How difficult is POS tagging in English?

Roughly 15% of word types are ambiguous

- Hence 85% of word types are unambiguous
- *Janet* is always PROPN, *hesitantly* is always ADV

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

Source: Jurafsky & Martin, SLP3 slides

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

Source: Jurafsky & Martin, SLP3 slides

Named Entity Recognition

Named Entities

- **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"
- Often multi-word phrases
- But the term is also extended to things that aren't entities:
  - dates, times, prices

Source: Jurafsky & Martin, SLP3 slides

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**].

Source: Jurafsky & Martin, SLP3 slides

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.

Source: Jurafsky & Martin, SLP3 slides

BIO Tagging

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

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

Now we have one tag per token!!!

Source: Jurafsky & Martin, SLP3 slides

BIO Tagging

B: token that *begins* a span

I: tokens *inside* a span

O: tokens outside of any span

# of tags (where n is #entity types):

1 O tag,

n B tags,

n I tags

total of *2n+1*

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

Source: Jurafsky & Martin, SLP3 slides

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

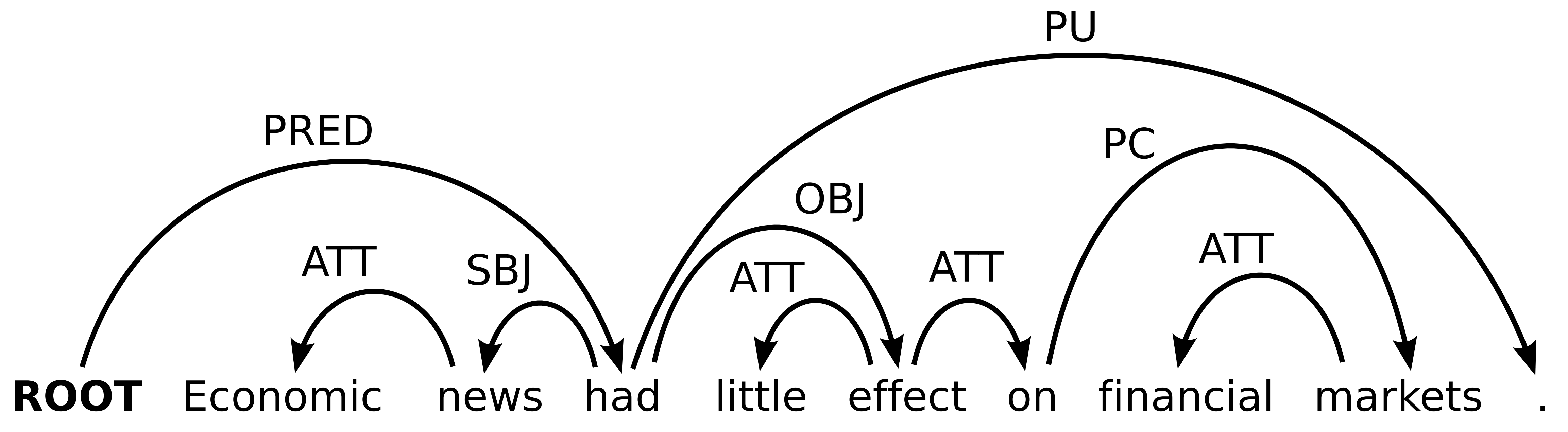
Source: Jurafsky & Martin, SLP3 slides

One common traditional approach to sequence labeling is "maximum entropy" modeling.

Pssst! Hot tip! “Maximum entropy” = “logistic regression”

Dependency parsing (and “universal dependency parsing”)

A dependency parse



Dependencies are (labeled) asymmetrical binary relations between two lexical items (words).

had —OBJ—> *effect* [*effect* is the object of *had*]

effect —ATT—> *little* [*little* is an attribute of *effect*]

We typically assume a special ROOT token as word 0

Source: Julia Hockenmaier, Illinois CS447 slides

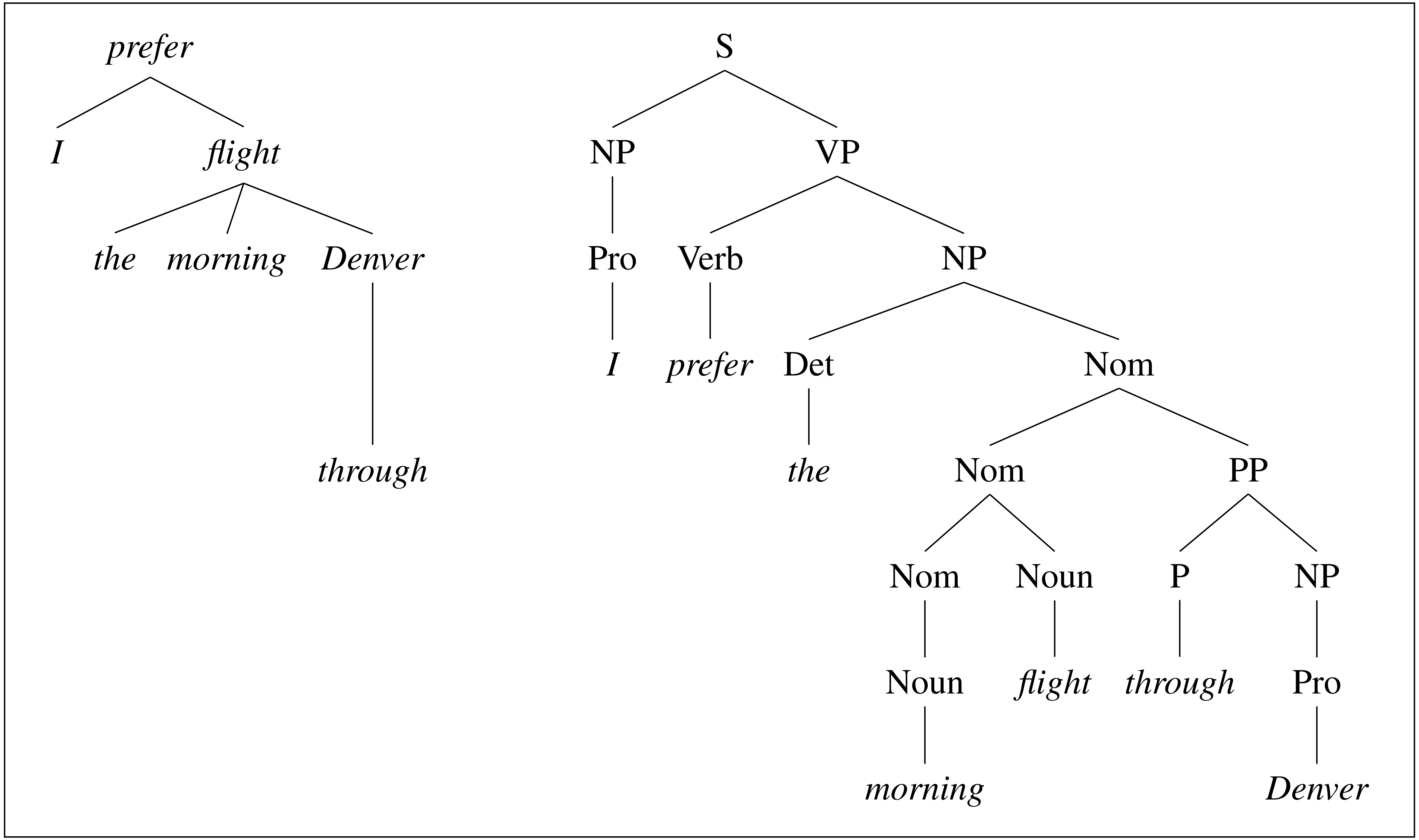


Figure 14.1 A dependency-style parse alongside the corresponding constituent-based analysis for *I prefer the morning flight through Denver.*

Figure source: Jurafsky and Martin, *SLP3*, 2021.

The popularity of Dependency Parsing

Currently the main paradigm for syntactic parsing.

Dependencies are easier to use and interpret
for downstream tasks than phrase-structure trees.

For languages with free word order, dependencies
are more natural than phrase-structure grammars

Dependency treebanks exist for many languages.

- The Universal Dependencies project has dependency treebanks for dozens of languages that use a similar annotation standard.

Source: Julia Hockenmaier, Illinois CS447 slides

Dependency grammar

Word-word dependencies are a component of many (most/all?) grammar formalisms.

Dependency grammar assumes that syntactic structure consists *only* of dependencies.

Many variants. Modern DG began with Tesniere (1959).

DG is often used for **free word order languages**.

DG is **purely descriptive** (not generative like CFGs etc.), but some formal equivalences are known.

Source: Julia Hockenmaier, Illinois CS447 slides

Dependency trees

Dependencies form a graph over the words in a sentence.

This graph is **connected** (every word is a node) and (typically) **acyclic** (no loops).

Single-head constraint:
Every node has at most **one incoming edge (each word has one parent)**

That means we can describe the parse tree of a sentence with one tag per token (its parent, or "root").

Together with connectedness, this implies that the graph is a **rooted tree**.

Source: Julia Hockenmaier, Illinois CS447 slides

Different kinds of dependencies

Head-argument: *eat sushi*

Arguments may be obligatory, but can only occur once.
The head alone cannot necessarily replace the construction.

Head-modifier: *fresh sushi*

Modifiers are optional, and can occur more than once.
The head alone can replace the entire construction.

Head-specifier: *the sushi*

Between function words (e.g. prepositions, determiners)
and their arguments. Here, syntactic head ≠ semantic head

?

Coordination: *sushi and sashimi*

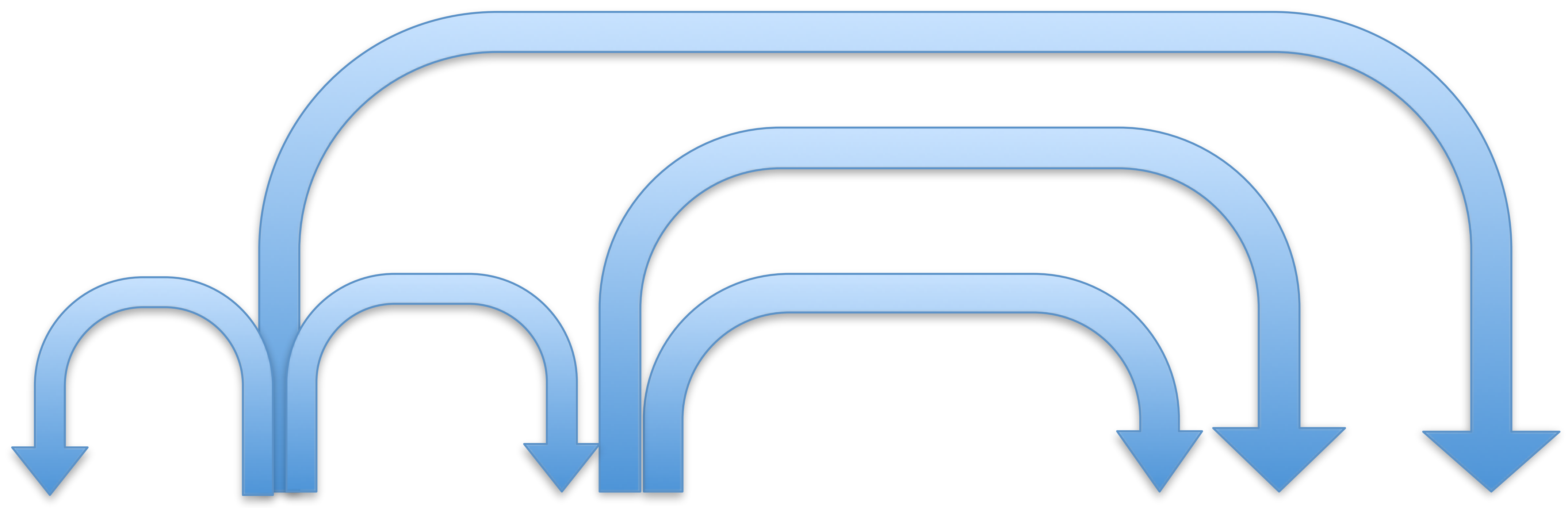
Unclear where the head is.

Source: Julia Hockenmaier, Illinois CS447 slides

Context-free grammars

CFGs capture only **nested** dependencies

- The dependency graph is a **tree**
- The dependencies **do not cross**



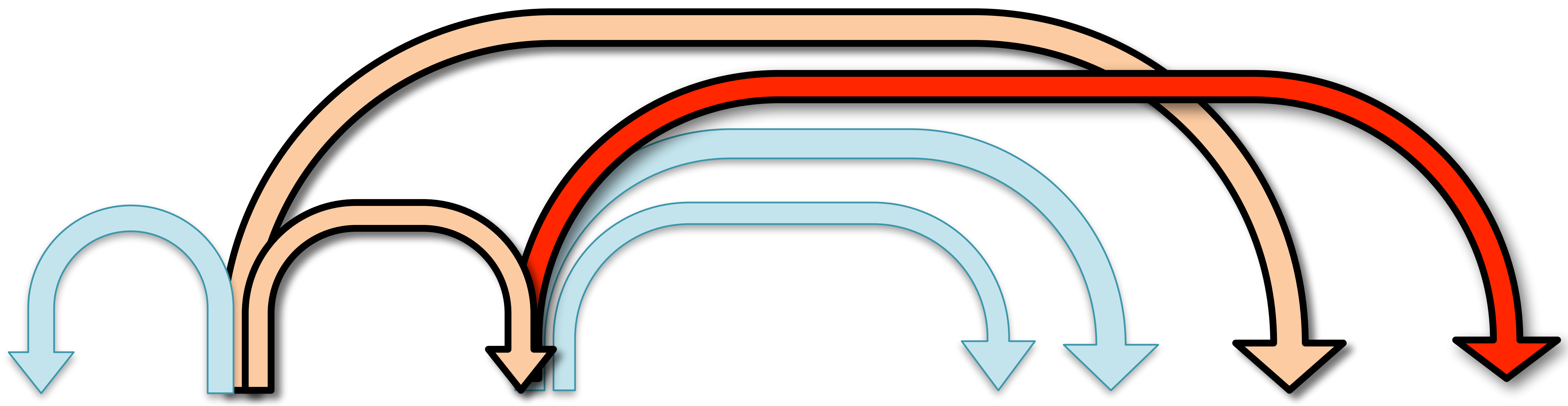
Source: Julia Hockenmaier, Illinois CS447 slides

Beyond CFGs: Nonprojective dependencies

Dependencies: tree with crossing branches

Arise in the following constructions

- (Non-local) **scrambling** (free word order languages)
  Die Pizza hat Klaus versprochen zu bringen
- **Extraposition** (*The guy is coming who is wearing a hat*)
- **Topicalization** (*Cheeseburgers, I thought he likes*)



Source: Julia Hockenmaier, Illinois CS447 slides

Dependency Treebanks

Dependency treebanks exist for many languages:

Czech
Arabic
Turkish
Danish
Portuguese
Estonian
....

Phrase-structure treebanks (e.g. the Penn Treebank) can also be translated into dependency trees (although there might be noise in the translation)

Source: Julia Hockenmaier, Illinois CS447 slides

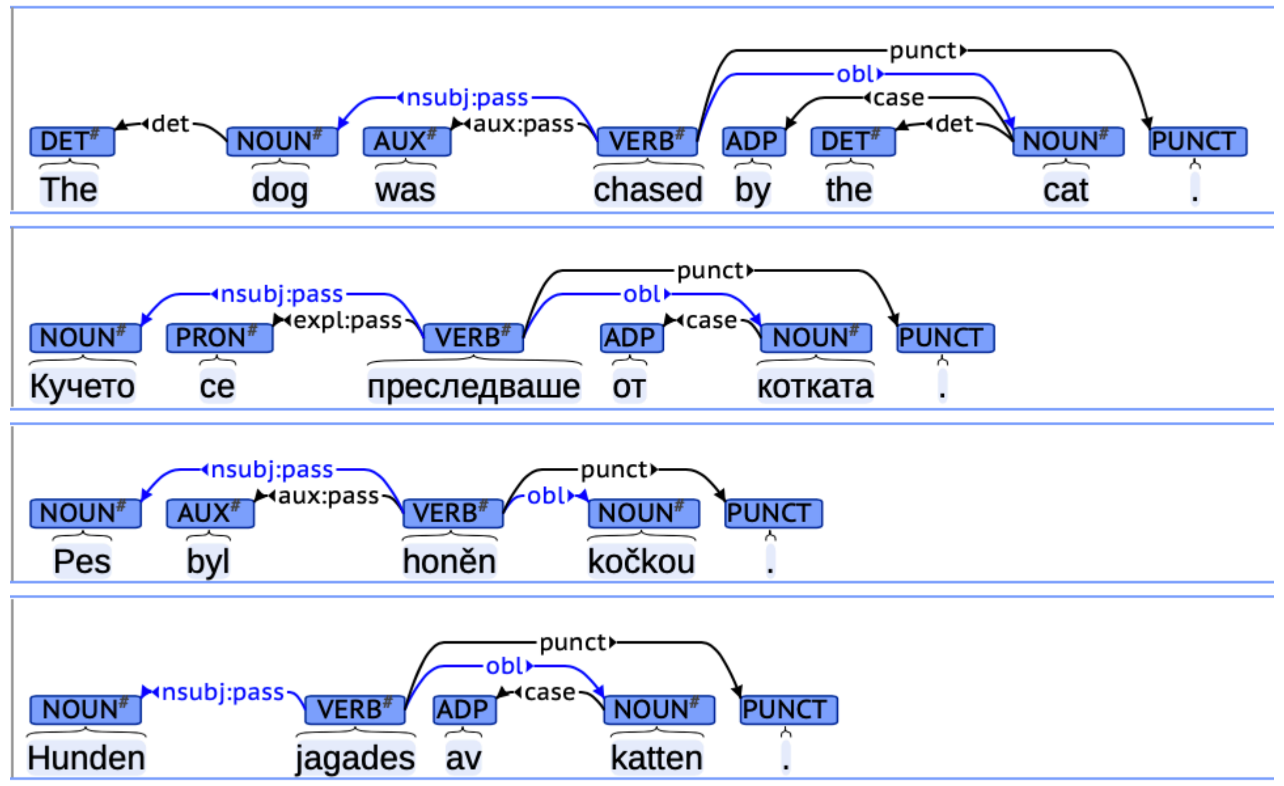
Universal Dependencies

37 syntactic relations, intended to be applicable to all languages (“universal”), with slight modifications for each specific language, if necessary.

http://universaldependencies.org

Example: “*the dog was chased by the cat*”
in English, Bulgarian, Czech and Swedish:

All languages have dependencies corresponding to
(*chased*, nsubj-pass, *dog*)
(*chased*, obj, *cat*)



Source: Julia Hockenmaier, Illinois CS447 slides

Universal Dependency Relations

Nominal core arguments: `nsubj` (nominal subject, incl. `nsubj-pass` (nominal subject in passive), `obj` (direct object), `iobj` (indirect object)

Clausal core arguments: `csubj` (clausal subject), `ccomp` (clausal object [“complement”])

Non-core (“oblique”) dependents: `obl` (oblique nominal argument or adjunct, e.g. for tools etc.), `advcl` (adverbial clause modifier), `aux` (auxiliary verb), `cop` (copula), `det` (determiner)

Nominal dependents: `nmod` (nominal modifier), `amod` (adjectival modifier), `appos` (appositional modifier)

Function words: `case` (case markers, prepositions), `det` (determiners),

Coordination: `cc` (coordinating conjunction), `conj` (conjunct)

Multiword Expressions: `compound` (within compound nouns), `flat` (dates, complex names, etc.),
Other: `root` (from ROOT to the head of the sentence), `dep` (catch-all label), `punct` (to punctuation marks)

Source: Julia Hockenmaier, Illinois CS447 slides

Relation	Examples with *head* and **dependent**
NSUBJ	**United** *canceled* the flight.
DOBJ	United *diverted* the **flight** to Reno.
	We *booked* her the first **flight** to Miami.
IOBJ	We *booked* **her** the flight to Miami.
NMOD	We took the **morning** *flight*.
AMOD	Book the **cheapest** *flight*.
NUMMOD	Before the storm JetBlue canceled **1000** *flights*.
APPOS	*United*, a **unit** of UAL, matched the fares.
DET	**The** *flight* was canceled.
	Which *flight* was delayed?
CONJ	We *flew* to Denver and **drove** to Steamboat.
CC	We flew to Denver **and** *drove* to Steamboat.
CASE	Book the flight **through** *Houston*.

Figure 14.3 Examples of core Universal Dependency relations.

Table source: Jurafsky and Martin, *SLP3*, 2021.

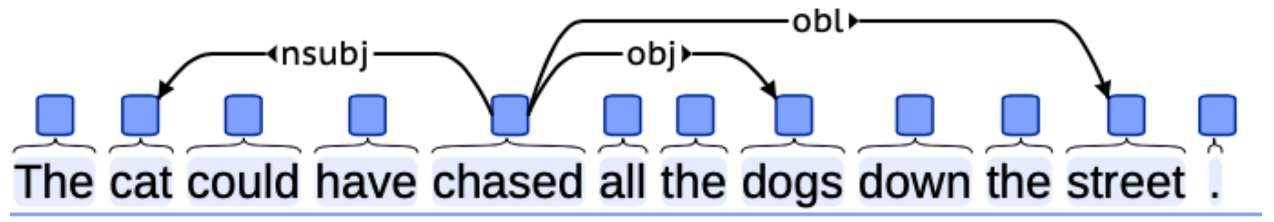
UD conventions: Primacy of content words

https://universaldependencies.org/u/overview/syntax.html

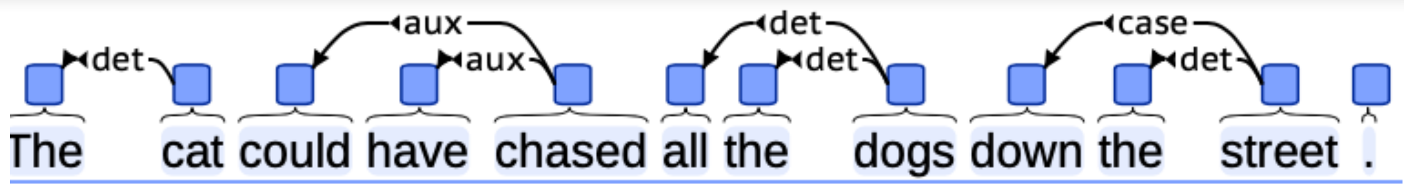
Dependency relations hold primarily between content words (which vary less across languages than function words)

Function words (prepositions, copulas, auxiliaries, determiners) attach to the most closely related content word, and typically don't have dependents

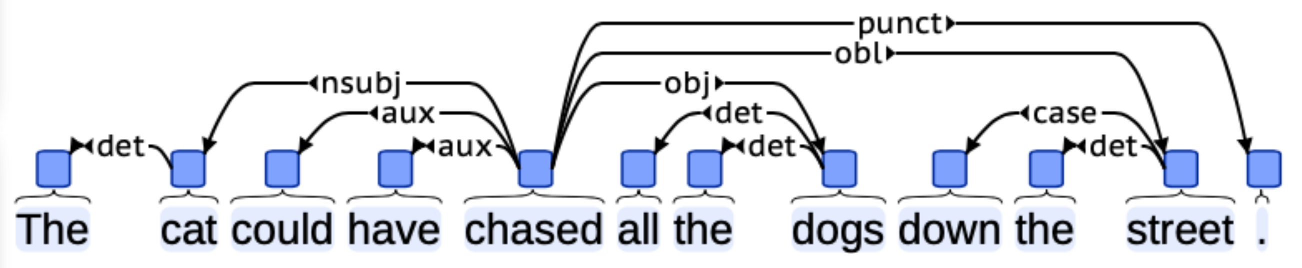
Content word dependencies



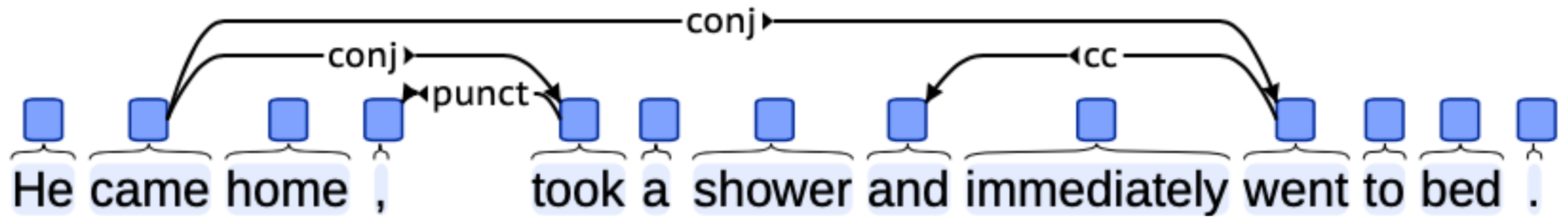
Function word dependencies



Complete analysis



In **coordination**, the first conjunct (*came*) is head, and the coordination (*and*) and subsequent conjuncts (*took, went*) depend on the first conjunct:



Source: Julia Hockenmaier, Illinois CS447 slides