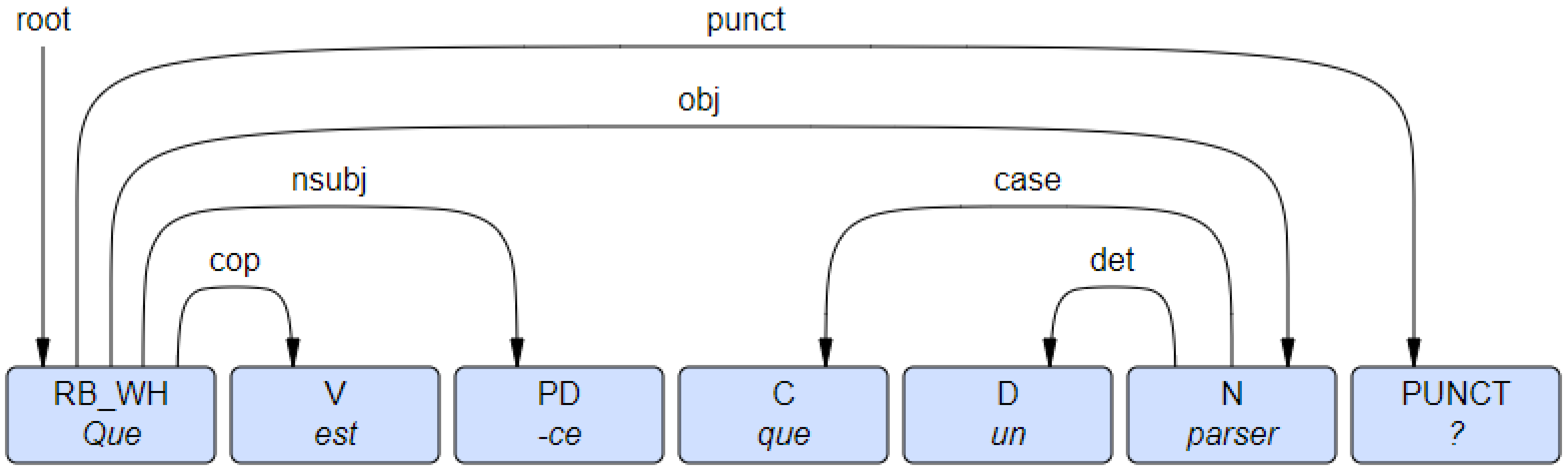
University of Ljubljana, Faculty of Computer and Information Science

Part-of-speech tagging, dependency parsing, and named entity recognition



Prof Dr Marko Robnik-Šikonja

Natural Language Processing, Version 2022

Contents

- POS tagging
- Tag sets
- Dependency parsing
- Universal dependencies
- Named entity recognition

Basic text processing

- document → paragraphs → sentences → words
- words and sentences ← **POS tagging**
- sentences ← **syntactical and grammatical analysis**

An Example

WORD	**LEMMA**	**TAG**
the	the	+DET
girl	girl	+NOUN
kissed	kiss	+VPAST
the	the	+DET
boy	boy	+NOUN
on	on	+PREP
the	the	+DET
cheek	cheek	+NOUN

First step: lemmatization

- Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item.
- Lemmatization difficulty is language dependent, i.e. it depends on morphology
- *English*
  - *walk, walked, walking, walks, ne pa walker*
  - *go, goes, going, gone, went*
- *Slovene*
  - *priti, pridem, prideš, pride, prideva, prideta, pridejo, pridemo, pridete, pridejo,* but not *prihod, prihodnost, prihajanje, prišlec*
  - *vlak, vlaka, vlaku, vlakom, vlakov,vlakoma,vlakih,vlaki, vlake*
  - *jaz, mene, meni, mano*
  - *Gori na gori gori!*
  - *Gori, na gori gori!*

Approaches to lemmatization

- Rules, dictionaries, lexicons, machine learning models
- Ambiguity resolution may be difficult

  Meni je vzel z mize (zapestnico). Zaradi vrata ni mogel odpreti vrat.
- Quick solutions and heuristics, in English just remove suffixes: *–ing, -ation, -ed, …*
- Essential approach for morphologically rich languages (Slavic, Arabic, Turkish, Spanish, etc)

Part-of-Speech Tagging

- Assigning a part-of-speech to each word in a text.
- Words often have more than one POS.
- **book**:
  - VERB: (***Book*** *that flight*)
  - NOUN: (*Hand me that* ***book***).

POS tagging

- Assigning the correct part of speech (noun, verb, etc.) to words
- Helps in recognizing phrases, names, terminology
- Helps in information retrieval, advanced search, named entity recognition, word sense disambiguation, coreference resolution, pronunciation, additional information for many classification tasks, useful heuristic for some tasks
- Helps in linguistic analyses such as verb valence, detection of multi-word expressions, semantic role labelling (SRL)

- Uses machine learning models

POS tagging for speech

- Speech synthesis:
  - How to pronounce “lead“? /liːd/ or /lɛd/

INsult	insult	noun: /ˈɪnsʌlt/	verb: /ɪnˈsʌlt/
OBject	obJECT		
OVERflow	overFLOW		
DIScount	disCOUNT		
CONtent	content		

- In Slovene
  - peti (to sing) peti (the fifth)

- Machine translation
  - The meaning of a particular word depends on its POS tag
- Sentiment analysis
  - Adjectives are the major opinion holders (good vs. bad, excellent vs. terrible)

Morphosyntactical tagging

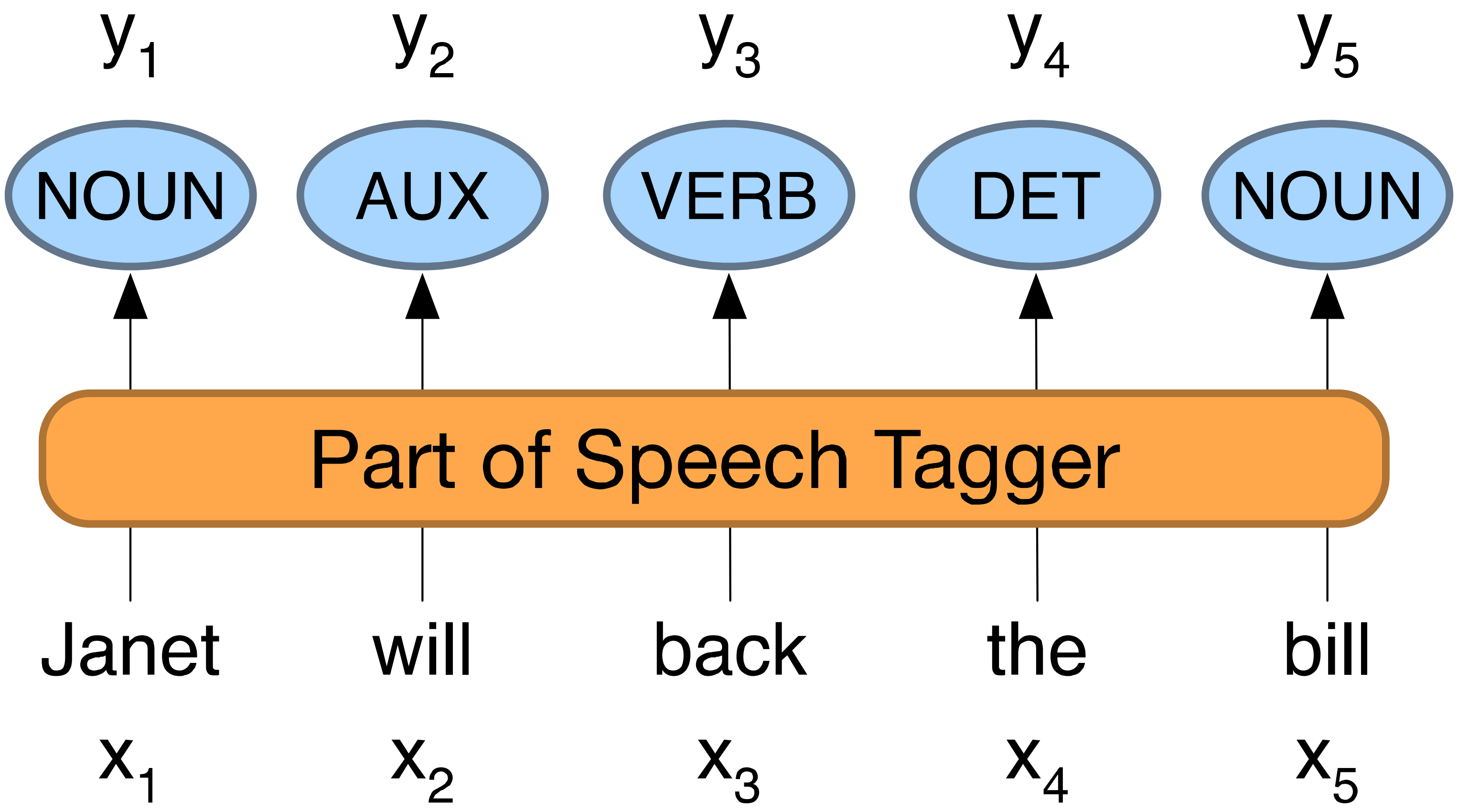
- POS tagging
- Basic categories from old Greek
  - noun, verb, pronoun, preposition, adjective/adverb, conjunction, participle, and article
  - samostalnik, glagol, zaimek, predlog, pridevnik/prislov, veznik, deležnik, členek
- Many additional features with important information: gender, tense, conjugation, etc.
- Tags defined based on
  - word morphology, e.g., suffixes and prefixes
  - distributional properties, i.e. neighborhood words, role in sentence
- Important part of disambiguation

POS 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

Part-of-Speech Tagging

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



Word classes: tag sets

- Vary in number of tags: for English from a dozen to over 200
- Size of tag sets depends on language, objectives and purpose
- We have to agree on a standard inventory of word classes
    - Taggers are trained on a labeled corpora
    - The tag set needs to capture semantically or syntactically important distinctions that can easily be made by trained human annotators

Open and closed class words

- Closed class: a relatively fixed membership
  - 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: new ones can be created all the time
  - 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.
  - New nouns and verbs like *iPhone* or *to fax*

Open class words

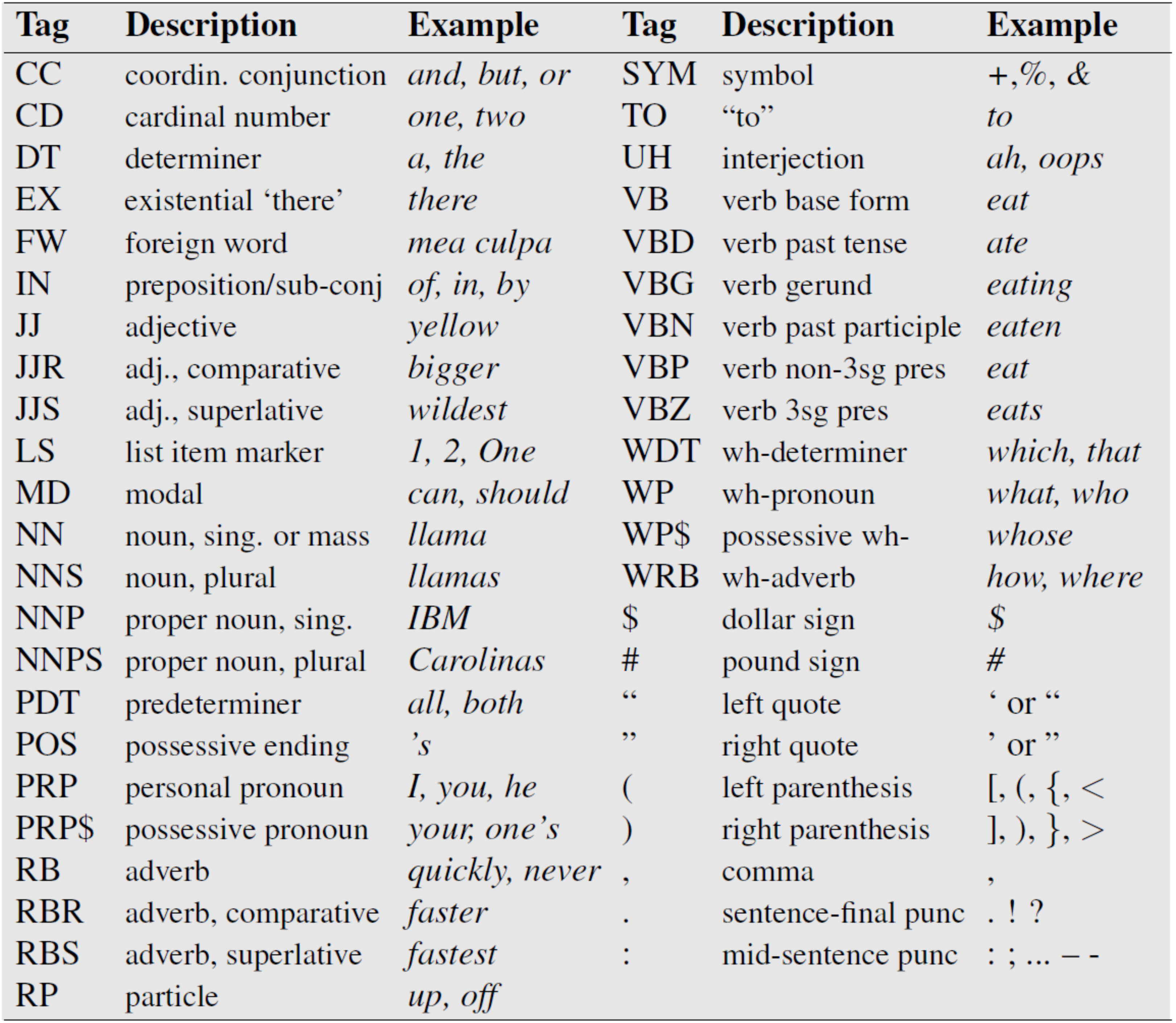
- Nouns
  - Proper nouns (Columbia University, New York City, Arthi Ramachandran, Metropolitan Transit Center). English capitalizes these.
  - Common nouns (the rest). German capitalizes these.
  - Count nouns and mass nouns
    - Count: have plurals, get counted: goat/goats, one goat, two goats
    - Mass: don't get counted (fish, salt, communism) (*two fishes refers to two species of fish)
- Adverbs: tend to modify things
  - Unfortunately, John walked home extremely slowly yesterday
  - Directional/locative adverbs (here, home, downhill)
  - Degree adverbs (extremely, very, somewhat)
  - Manner adverbs (slowly, slinkily, delicately)

Open class words

- Verbs:
  - In English, they have morphological affixes (eat/eats/eaten)
  - Actions (walk, ate) and states (be, exude)
  - Many subclasses, e.g.
    - eats/VBZ, eat/VB, eat/VBP, eats/VBZ, ate/VBD, eaten/VBN, eating/VBG, ...
    - Reflect morphological form & syntactic function

Tag set example

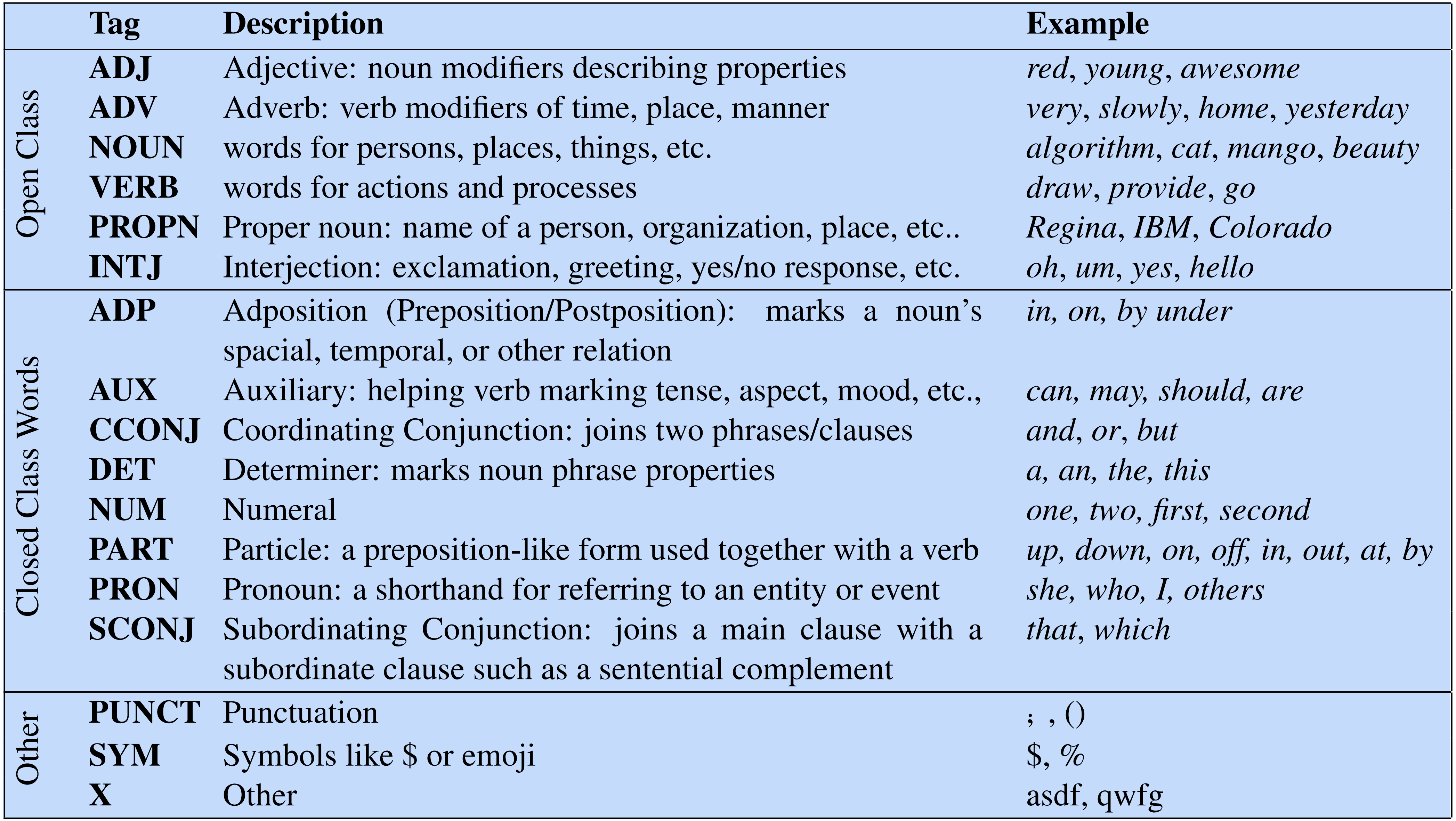
- e.g., Penn-Treebank tag set
- between 45 and 70 tags



Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two*	TO	“to”	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential ‘there’	*there*	VB	verb base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb gerund	*eating*
JJ	adjective	*yellow*	VBN	verb past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, sing.	*IBM*	$	dollar sign	$
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	“	left quote	‘ or “
POS	possessive ending	*’s*	”	right quote	’ or ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one’s*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

"Universal Dependencies" Tagset

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

Public tag sets in English

- Brown corpus - Francis and Kucera 1961
  - 500 samples, distributed across 15 genres in rough proportion to the amount published in 1961 in each of those genres
  - 87 tags
- Penn Treebank - Marcus et al. 1993
  - Hand-annotated corpus of Wall Street Journal, 1M words
  - 45 tags, a simplified version of Brown tag set
  - Standard for English now
    - Most statistical POS taggers are trained on this tagset
- Universal Dependencies (UD) – introduced later

Example of Penn Treebank Tagging of Brown Corpus Sentence

- The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

- VB   DT NN   .
Book that flight .

- VBZ DT   NN   VB   NN   ?
Does that flight serve dinner ?

The Problem

- Words often have more than one word class: *this*
  - *This* is a nice day = PRP (personal pronoun)
  - *This* day is nice = DT (determiner)
  - You can go *this* far = RB (adverb)
- *Back*
  - The *back* door (adjective)
  - On my *back* (noun)
  - Promised to *back* the bill (verb)

Buffalo example

- A grammatically correct (but lexically ambiguous) sentence in American English: **Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo.**
- Dmitri Borgmann, 1967. *Beyond Language: Adventures in Word and Thought*.
- The sentence employs three distinct meanings of the word *buffalo*:
  - as a proper noun to refer to a specific place named Buffalo, the city of Buffalo, New York, being the most notable;
  - as a verb (uncommon in regular usage) *to buffalo*, meaning "to bully, harass, or intimidate" or "to baffle"; and
  - as a noun to refer to the animal, bison (often called *buffalo* in North America). The plural is also *buffalo*.
- An expanded form of the sentence which preserves the original word order is: "Buffalo bison, that other Buffalo bison bully, also bully Buffalo bison."

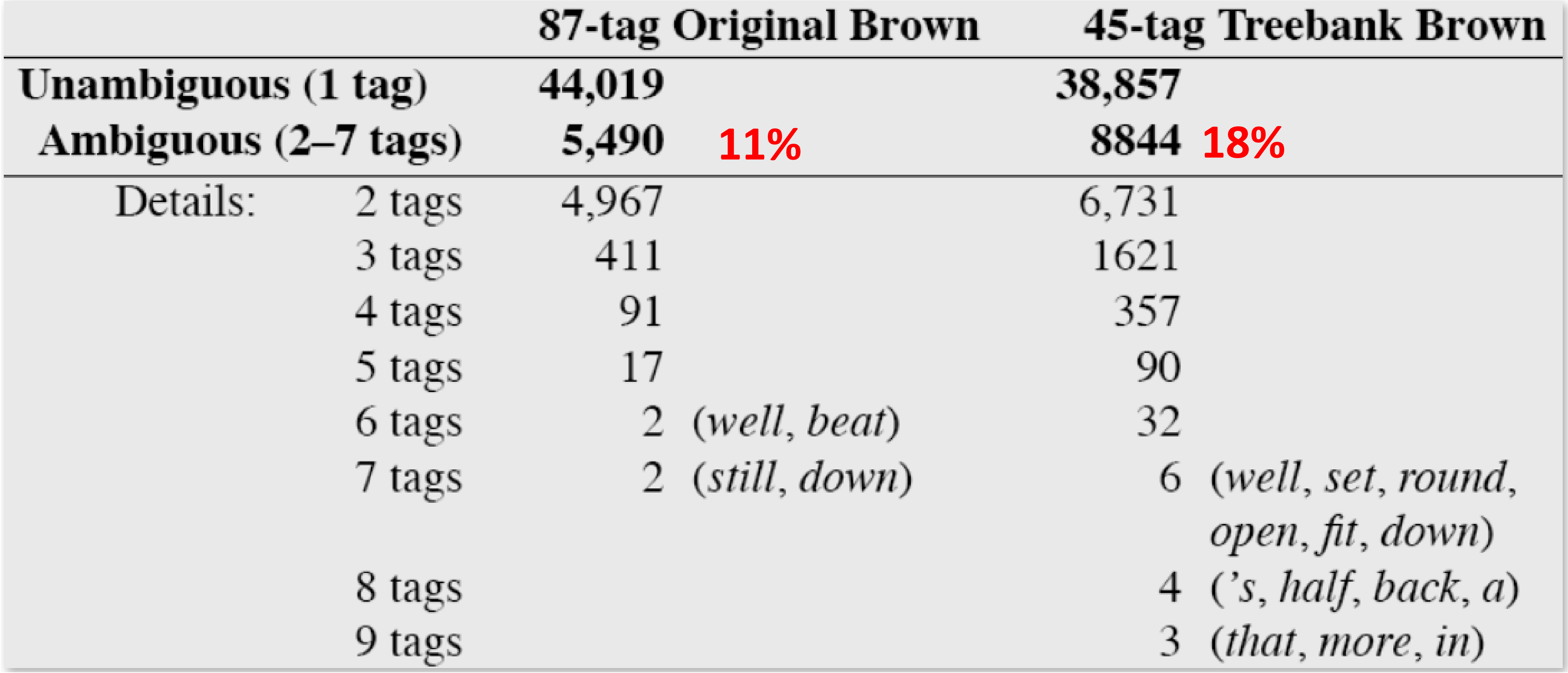
How difficult is POS tagging in English?

- 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

How much ambiguity is there?

- Statistics of word-tag pair in Brown Corpus and Penn Treebank



		87-tag Original Brown		45-tag Treebank Brown	
Unambiguous (1 tag)		**44,019**		**38,857**	
Ambiguous (2–7 tags)		**5,490**	11%	**8844**	18%
Details:	2 tags	4,967		6,731	
	3 tags	411		1621	
	4 tags	91		357	
	5 tags	17		90	
	6 tags	2	(*well*, *beat*)	32	
	7 tags	2	(*still*, *down*)	6	(*well*, *set*, *round*, *open*, *fit*, *down*)
	8 tags			4	(*’s*, *half*, *back*, *a*)
	9 tags			3	(*that*, *more*, *in*)

POS tagging baselines

- Default classifier:
  - each word is assigned the most probable category,
  - probabilities are computed from manually tagged corpus,
  - in English around 92% classification accuracy
- Human expert accuracy is around 98%

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

Is POS tagging a solved problem?

- Baseline
  - Tag every word with its most frequent tag
  - Tag unknown words as nouns
- Accuracy
  - Word level: 90%
  - Sentence level
    - Average English sentence length 14.3 words
    - $0.9^{14.3} = 22\%$

Accuracy of better POS Tagger

- *Word level: 97%*
- *Sentence level:* $0.97^{14.3} = 65\%$

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 AUX
- 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 → PROPN

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

Classical ML models

- SVM
- Conditional Random Fields (CRF)

- Approach:
  - define a set of useful features
  - train a ML model
- Let us illustrate this approach on Slovene

Morphosyntactical tagging for Slovene

- Slovene is morphologically rich language
- Large set of tags (1902 tags), why?
- Free word order means that certain taggers do not work well, e.g., HMM
- History of tagging
  - MULTEXT-East
    - Around 100.000 words
    - Very homogenous source, a single novel (George Orwell: 1984)
  - JOS 100k / 1M
    - Around 100.000 / 1.000.000 words
    - More heterogeneous
    - Manually labelled 100k corpus / corpus of 1M words partially manually labelled (estimate: 96%accurate tags)
    - Based on FidaPLUS corpus containing 620 million words

Current Slovene POS dataset

- ssj500k
  - Currently 600k manually labelled corpus
  - Being extended to 1M
  - Analysis of common errors (mostly due to underrepresentation of certain tags in the corpus), e.g., je

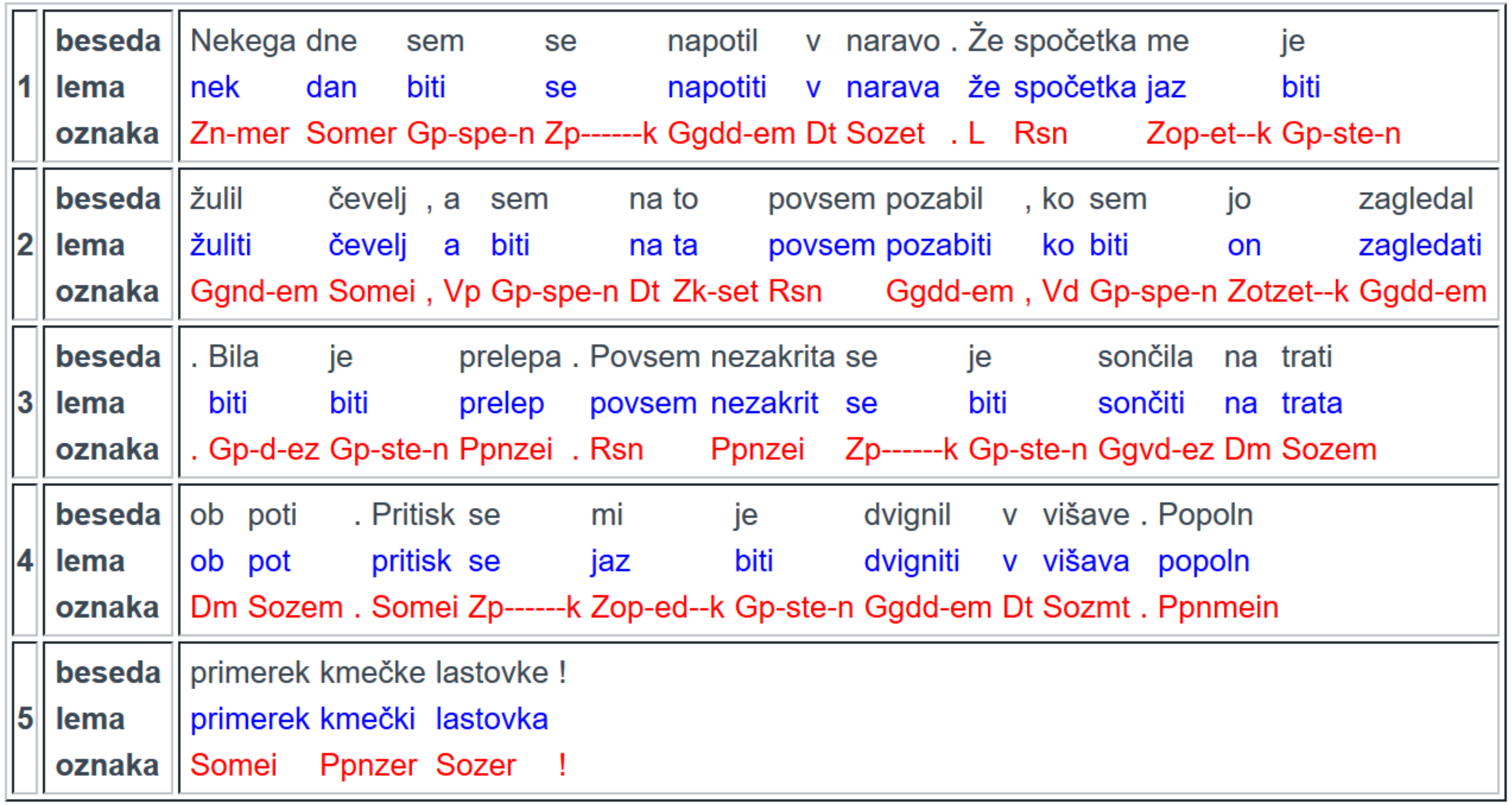
An example in Slovene

- JOS ToTaLe text analyzer for Slovene: morphosyntactical tagging, (old variant available at http://www.slovenscina.eu/)

  Nekega dne sem se napotil v naravo. Že spočetka me je žulil čevelj, a sem na to povsem pozabil, ko sem jo zagledal. Bila je prelepa. Povsem nezakrita se je sončila na trati ob poti. Pritisk se mi je dvignil v višave. Popoln primerek kmečke lastovke!
- Tags are standardized for East European languages in Multext-East specification, e.g.,

dne; tag Somer = Samostalnik, obče ime, moški spol, ednina, rodilnik; lema: dan

- *Nekega dne sem se napotil v naravo. Že spočetka me je žulil čevelj, a sem na to povsem pozabil, ko sem jo zagledal. Bila je prelepa. Povsem nezakrita se je sončila na trati ob poti. Pritisk se mi je dvignil v višave. Popoln primerek kmečke lastovke!*



1	**beseda**	Nekega	dne	sem	se	napotil	v	naravo	.	Že	spočetka	me	je
	lema	nek	dan	biti	se	napotiti	v	narava		že	spočetka	jaz	biti
	oznaka	Zn-mer	Somer	Gp-spe-n	Zp------k	Ggdd-em	Dt	Sozet	.	L	Rsn	Zop-et--k	Gp-ste-n

2	**beseda**	žulil	čevelj	,	a	sem	na	to	povsem	pozabil	,	ko	sem	jo	zagledal
	lema	žuliti	čevelj		a	biti	na	ta	povsem	pozabiti		ko	biti	on	zagledati
	oznaka	Ggnd-em	Somei	,	Vp	Gp-spe-n	Dt	Zk-set	Rsn	Ggdd-em	,	Vd	Gp-spe-n	Zotzet--k	Ggdd-em

3	**beseda**	.	Bila	je	prelepa	.	Povsem	nezakrita	se	je	sončila	na	trati
	lema		biti	biti	prelep		povsem	nezakrit	se	biti	sončiti	na	trata
	oznaka	.	Gp-d-ez	Gp-ste-n	Ppnzei	.	Rsn	Ppnzei	Zp------k	Gp-ste-n	Ggvd-ez	Dm	Sozem

4	**beseda**	ob	poti	.	Pritisk	se	mi	je	dvignil	v	višave	.	Popoln
	lema	ob	pot		pritisk	se	jaz	biti	dvigniti	v	višava		popoln
	oznaka	Dm	Sozem	.	Somei	Zp------k	Zop-ed--k	Gp-ste-n	Ggdd-em	Dt	Sozmt	.	Ppnmein

5	**beseda**	primerek	kmečke	lastovke	!
	lema	primerek	kmečki	lastovka	
	oznaka	Somei	Ppnzer	Sozer	!

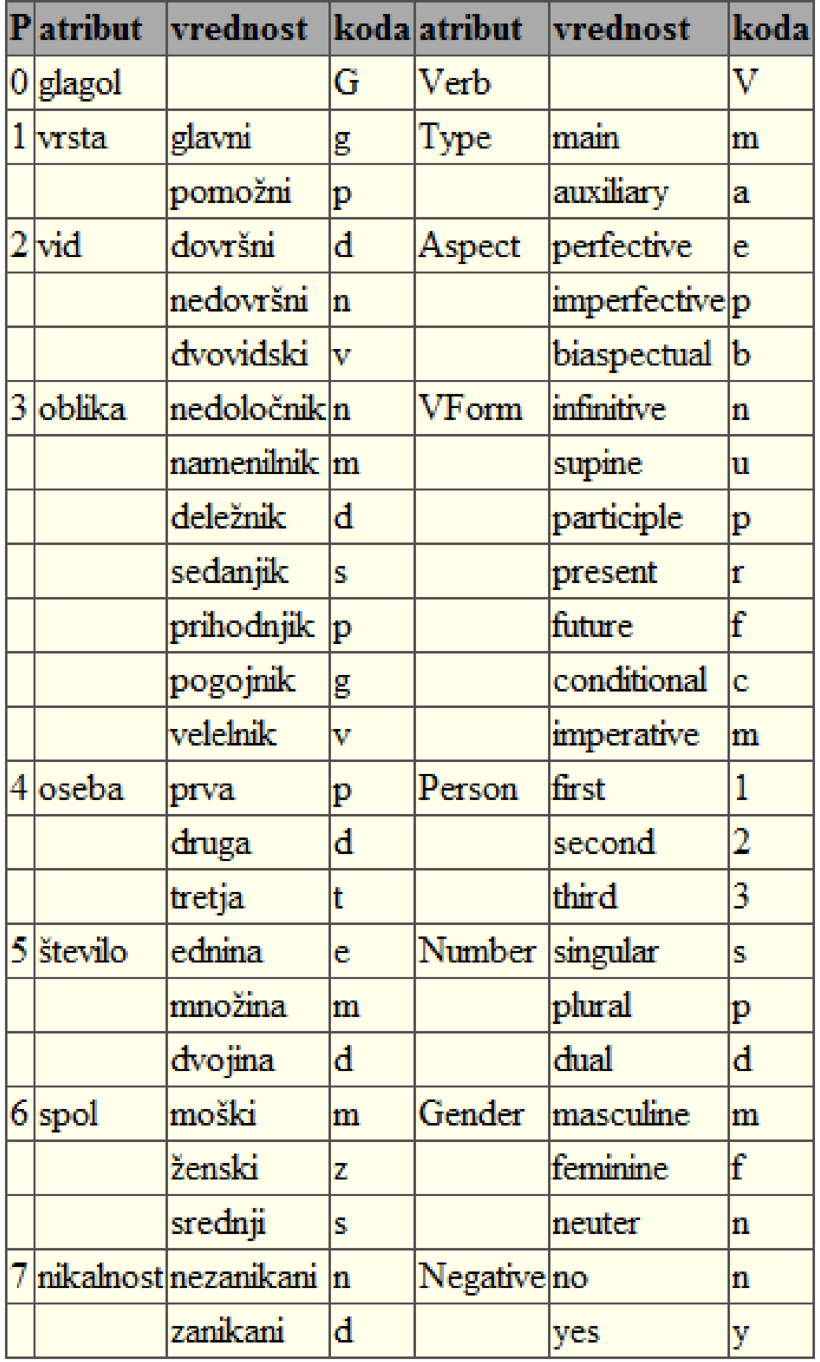
TEI-XML format

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <text>
    <body>
      <p>
        <s>
          <w msd="Zn-mer" lemma="nek">Nekega</w>
          <S/>
          <w msd="Somer" lemma="dan">dne</w>
          <S/>
          <w msd="Gp-spe-n" lemma="biti">sem</w>
          <S/>
          <w msd="Zp------k" lemma="se">se</w>
          <S/>
          <w msd="Ggdd-em" lemma="napotiti">napotil</w>
          <S/>
          <w msd="Dt" lemma="v">v</w>
          <S/>
          <w msd="Sozet" lemma="narava">naravo</w>
          <c>.</c>
          <S/>
        </s>
   …

      </p>
    </body>
  </text>
</TEI>
```

MSD tags for Slovene

- Multext-East 4.0 specification
- example: dne;
  tag Somer = Samostalnik, obče ime, moški spol, ednina, rodilnik; lema: dan
- below top level tags there are many informative features
- example for verb



P	atribut	vrednost	koda	atribut	vrednost	koda
0	glagol		G	Verb		V
1	vrsta	glavni	g	Type	main	m
		pomožni	p		auxiliary	a
2	vid	dovršni	d	Aspect	perfective	e
		nedovršni	n		imperfective	p
		dvovidski	v		biaspectual	b
3	oblika	nedoločnik	n	VForm	infinitive	n
		namenilnik	m		supine	u
		deležnik	d		participle	p
		sedanjik	s		present	r
		prihodnjik	p		future	f
		pogojnik	g		conditional	c
		velelnik	v		imperative	m
4	oseba	prva	p	Person	first	1
		druga	d		second	2
		tretja	t		third	3
5	število	ednina	e	Number	singular	s
		množina	m		plural	p
		dvojina	d		dual	d
6	spol	moški	m	Gender	masculine	m
		ženski	z		feminine	f
		srednji	s		neuter	n
7	nikalnost	nezanikani	n	Negative	no	n
		zanikani	d		yes	y

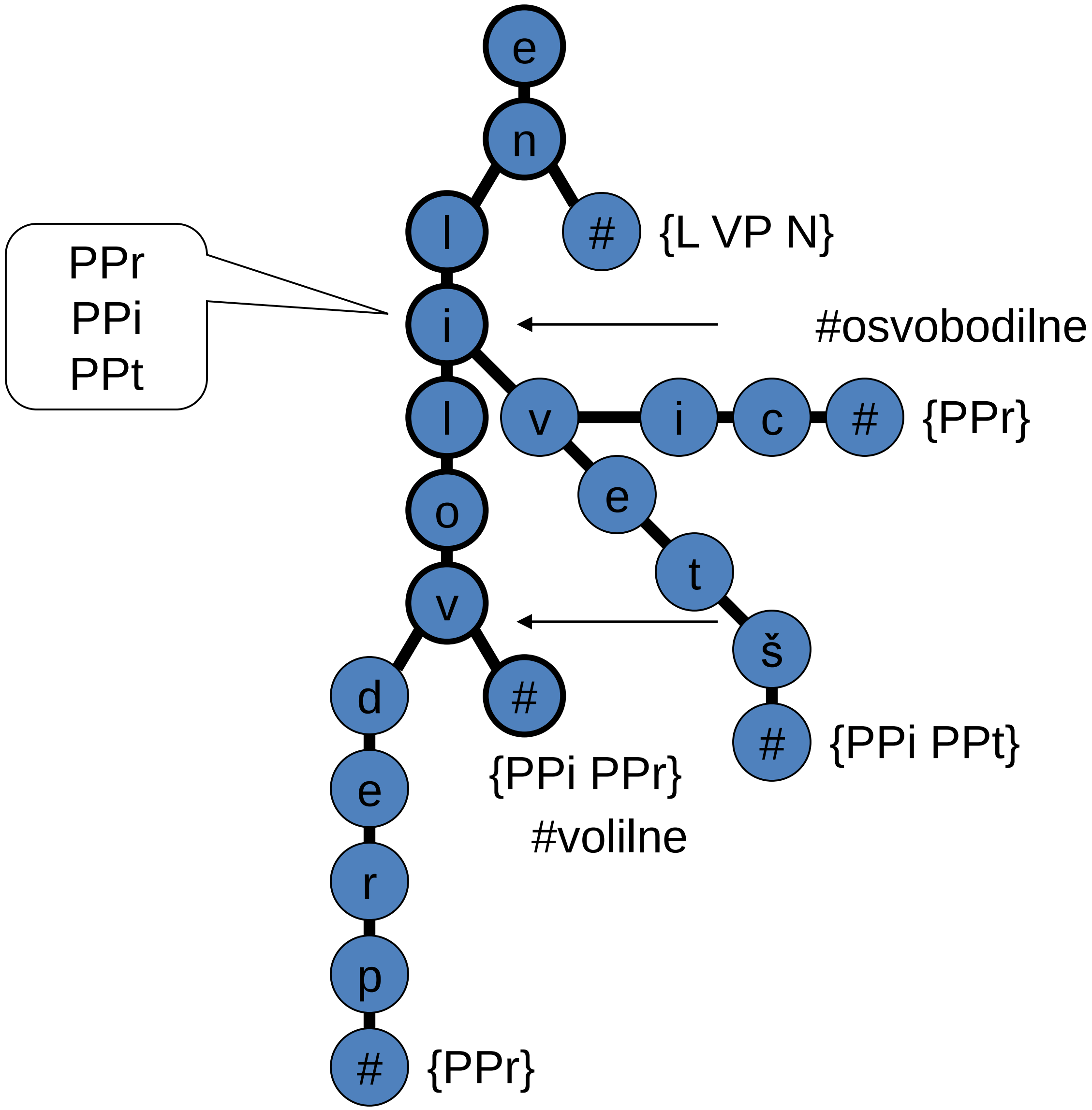
Example: Slovene Obeliks tagger

- slides taken from

Miha Grčar: Oblikoskladenjski označevalnik SSJ, presented at conference Korpusi, več kot le statistika (Fakulteta za družbene vede, Ljubljana, 5. februar 2010)

- Obeliks uses machine learning from manually labelled examples

Suffix trie


e
n
l
#
{L VP N}
PPr
PPi
PPt
i
#osvobodilne
l
v
i
c
#
{PPr}
o
e
t
v
š
d
#
#
{PPi PPt}
e
{PPi PPr}
#volilne
r
p
#
{PPr}

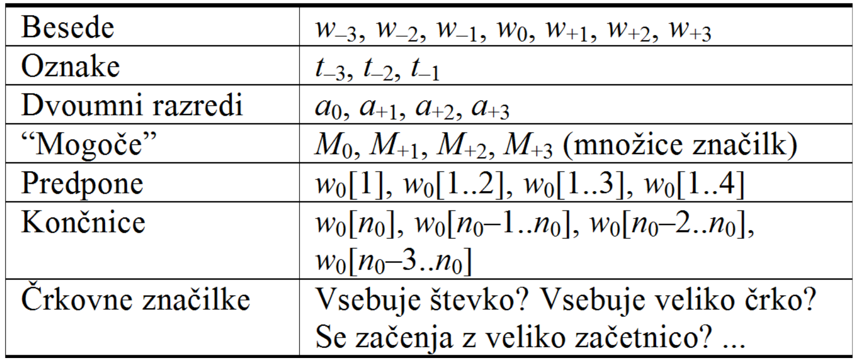
Features for ML

(Gimenez and Marquez, 2004)

L D P ?

Še v najboljših časih je redko delovalo, zdaj ...

- w_{-3}=še, w_{-2}=v, …, w_{+3}=delovalo
- t_{-3}=L, t_{-2}=D, t_{-1}=P
- a_0={ S }, a_{+1}={ G Z }, a_{+2}={ R }, a_{+3}={ P S G… }
- M_0=S, M_{+1}=G, M_{+1}=Z, …, M_{+3}=P, …M_{+3}=S, M_{+3}=G, …
- $w_0[1]$=č, $w_0[1..2]$=ča, ...
- $w_0[n_0]$=h, $w_0[n_0–1..n_0]$=ih, …
- contains number=no
- contains capital letter=no
- Starts with a capital letter=no …



Besede	$w_{-3}, w_{-2}, w_{-1}, w_0, w_{+1}, w_{+2}, w_{+3}$
Oznake	t_{-3}, t_{-2}, t_{-1}
Dvoumni razredi	$a_0, a_{+1}, a_{+2}, a_{+3}$
“Mogoče”	$M_0, M_{+1}, M_{+2}, M_{+3}$ (množice značilk)
Predpone	$w_0[1], w_0[1..2], w_0[1..3], w_0[1..4]$
Končnice	$w_0[n_0], w_0[n_0–1..n_0], w_0[n_0–2..n_0], w_0[n_0–3..n_0]$
Črkovne značilke	Vsebuje števko? Vsebuje veliko črko? Se začenja z veliko začetnico? ...

words

tags

sets of possible other tags

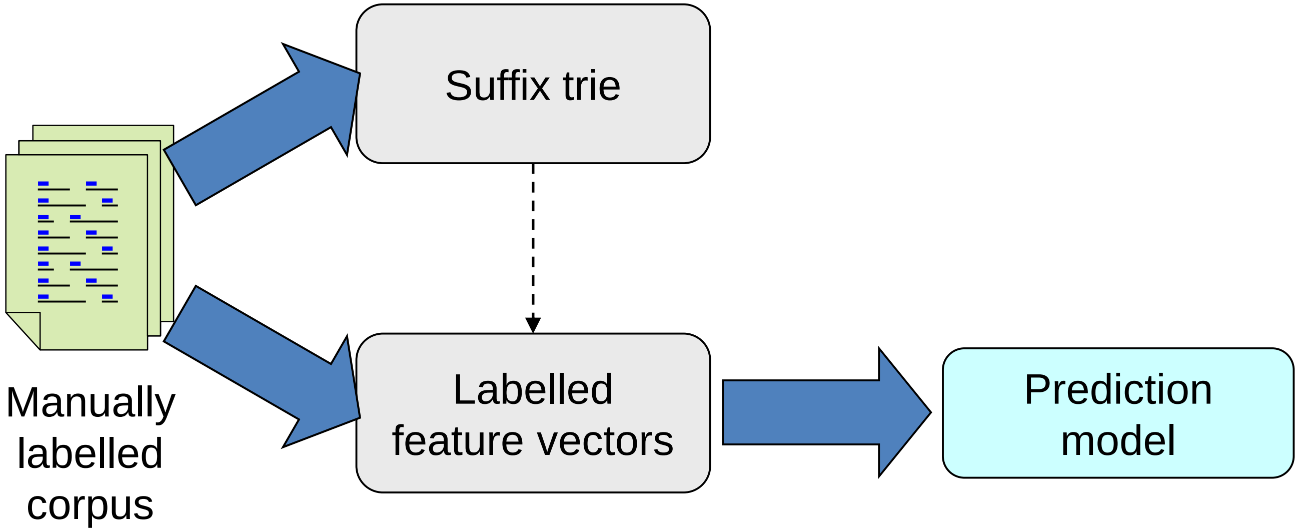
possible tags

prefixes

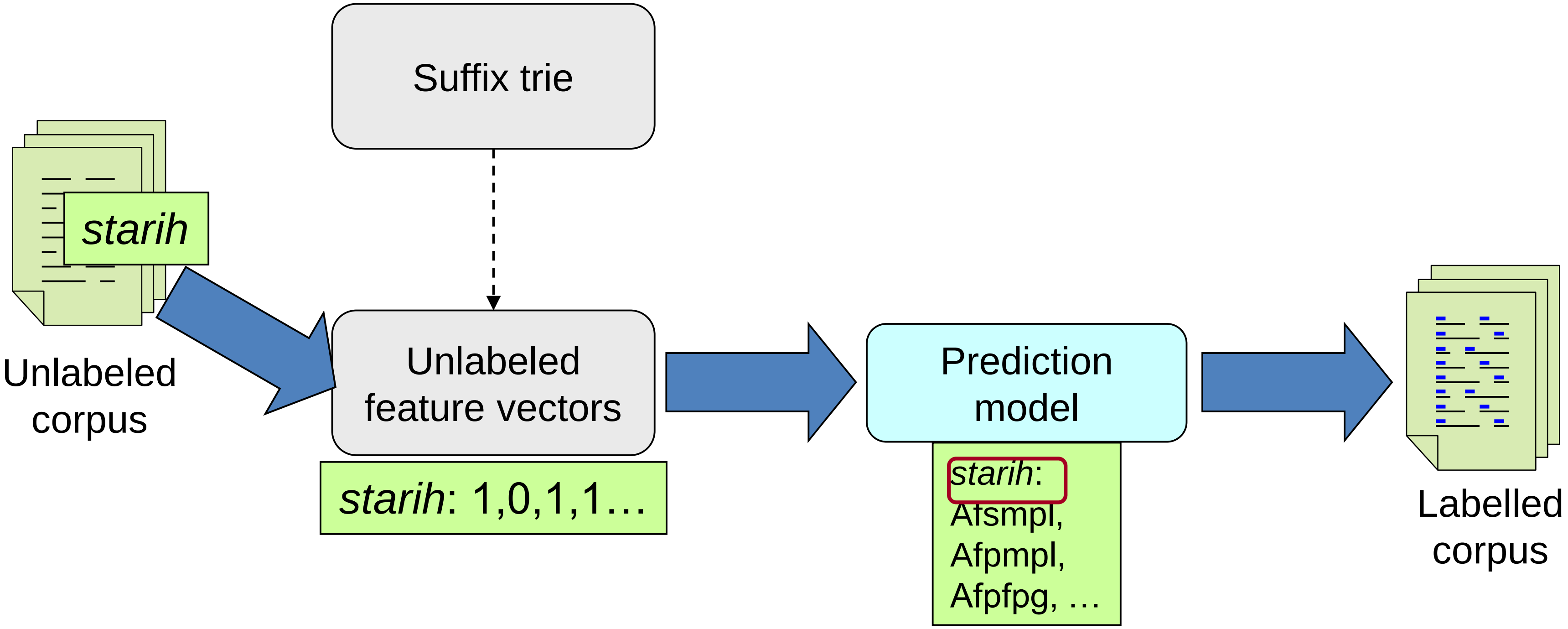
suffixes

letter based features

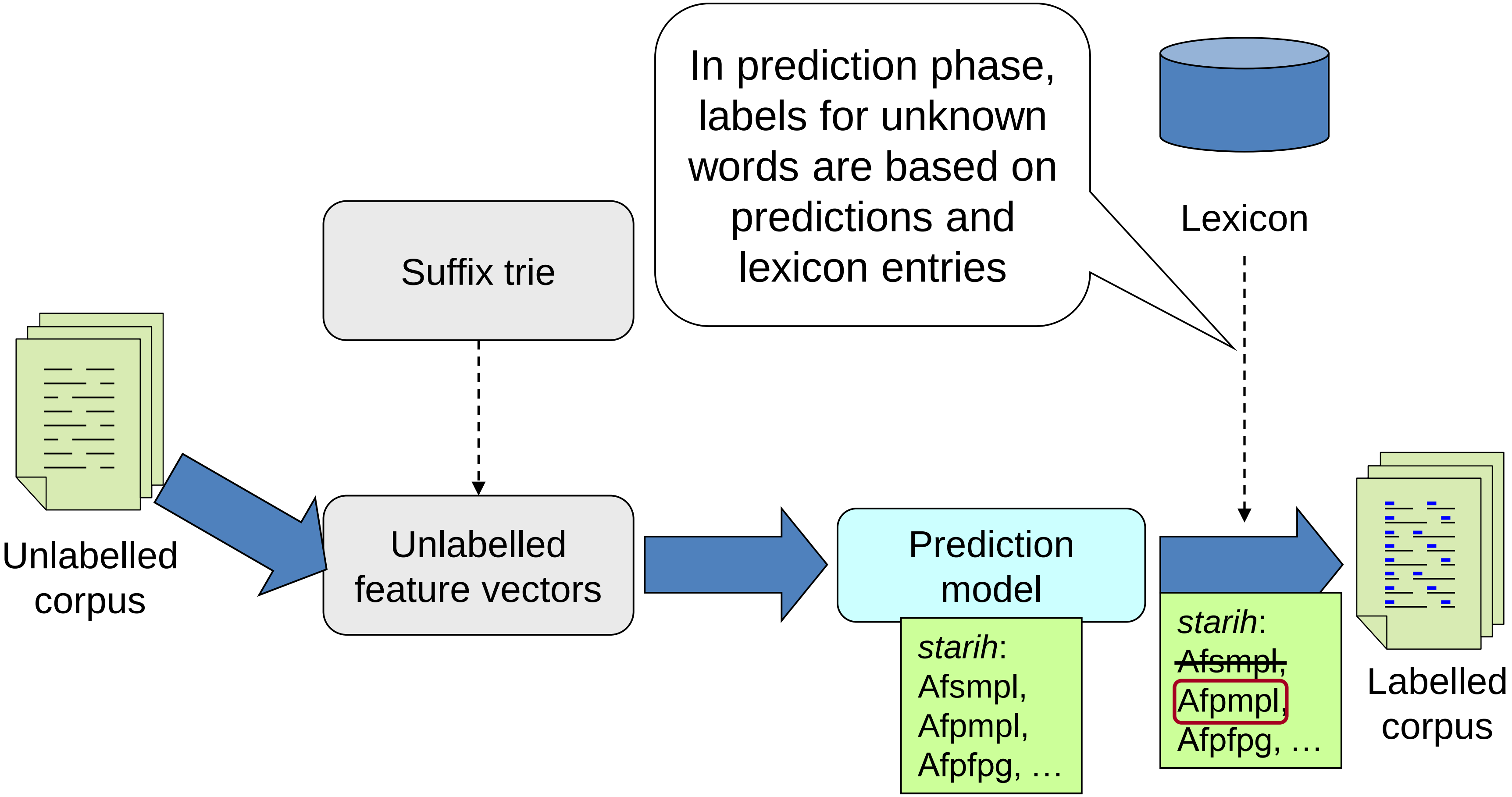
Training


Suffix trie
Manually labelled corpus
Labelled feature vectors
Prediction model

Prediction


Suffix trie
starih
Unlabeled corpus
Unlabeled feature vectors
starih: 1,0,1,1…
Prediction model
starih:
Afsmpl,
Afpmpl,
Afpfpg, …
Labelled corpus

Using lexicon in prediction



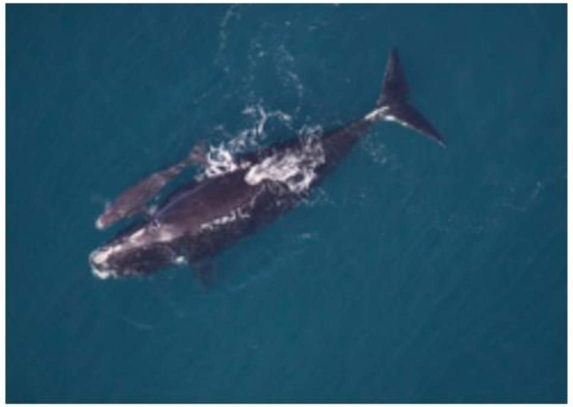
In prediction phase, labels for unknown words are based on predictions and lexicon entries
Lexicon
Suffix trie
Unlabelled corpus
Unlabelled feature vectors
Prediction model
starih: Afsmpl, Afpmpl, Afpfpg, …
starih: ~~Afsmpl,~~ Afpmpl, Afpfpg, …
Labelled corpus

Parsing: finding linguistic structure

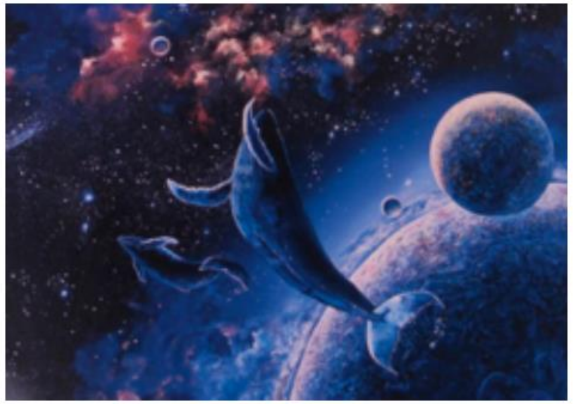
1. Constituency parsing
2. Dependency parsing

Parsing reduces ambiguity

Scientists count whales from space



Scientists count whales from space



Constituency parsing

- Dependency structure shows which words depend on (modify or are arguments of) which other words.
- *Look in the large crate in the kitchen by the door*
- We need to understand sentence structure in order to be able to interpret language correctly
- Humans communicate complex ideas by composing words together into bigger units to convey complex meanings
- We need to know what is connected to what

Constituency parsing

- Phrase structure organizes words into nested constituents
- Starting unit: words are given a category (part of speech = pos)

  the, cat, cuddly, by, door
- Words combine into phrases with categories

  the cuddly cat, by the door
- Phrases can combine into bigger phrases recursively

  the cuddly cat by the door

  Det Adj N P Det N
- Words combine into phrases with categories

  the cuddly cat, by the door

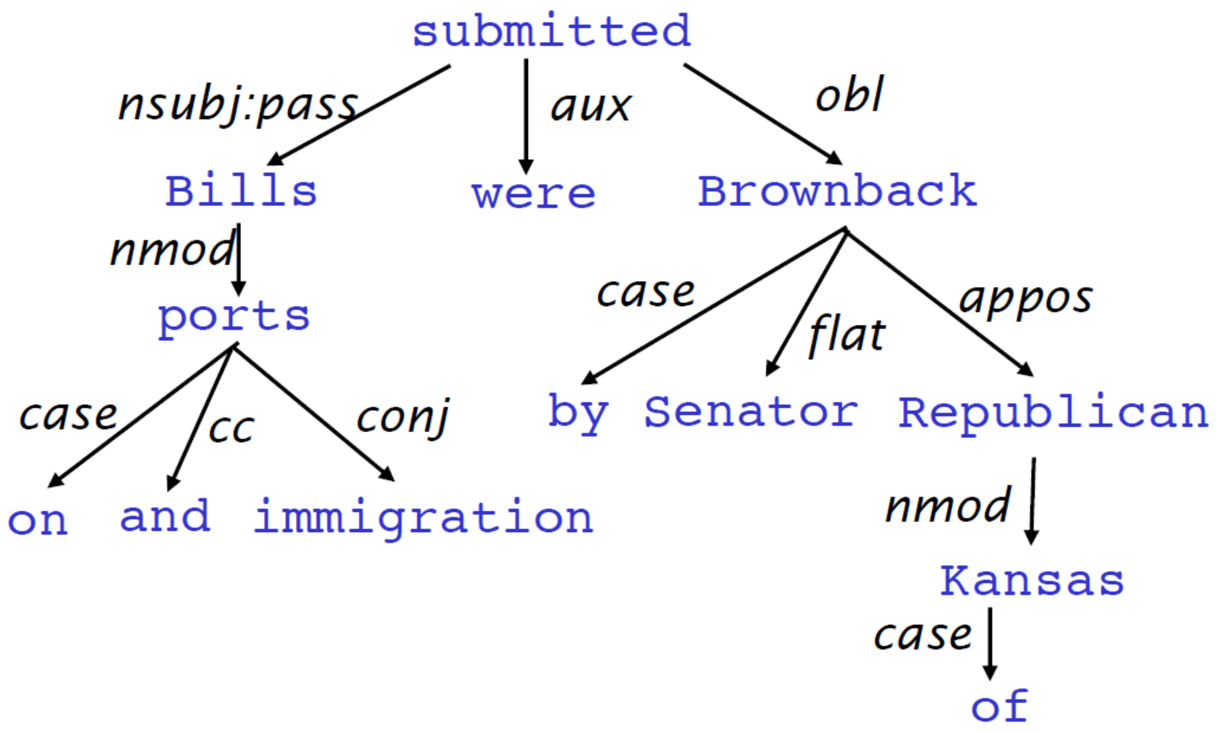
  NP →Det Adj N NP → Det N PP →P NP
- Phrases can combine into bigger phrases recursively

  the cuddly cat by the door NP →NP PP

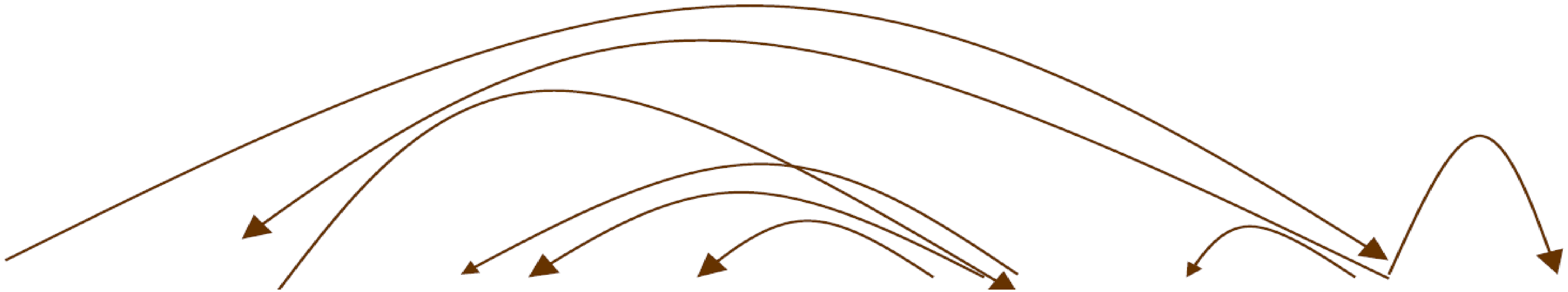
Dependency parsing

- Dependency syntax postulates that syntactic structure consists of relations between lexical items, normally binary asymmetric relations (“arrows”) called dependencies

The arrows are commonly typed with the name of grammatical relations (subject, prepositional object, apposition, etc.)



Dependency Grammar and Dependency Structure



ROOT Discussion of the outstanding issues was completed .

- Some people draw the arrows one way; some the other way!
- Usually add a fake ROOT so every word is a dependent of precisely 1 other node

Advantages of dependency parsing

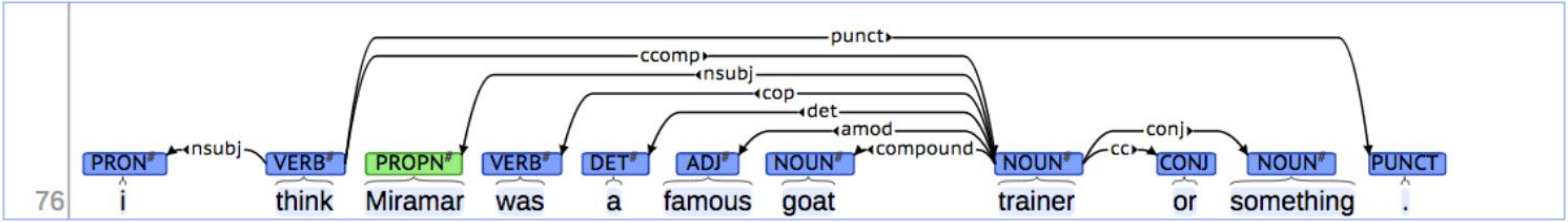
- Better handling of free word order (less-Anglo-centric)
- Node simplicity
- Clean mapping to semantic predicate-argument structure
- Easier to develop multilingual systems

Role of dependency parsing in NLP

- Semantic role labeling
- Relation extraction,
- Machine translation,
- Important role in the linguistic analysis

Treebanks

- The rise of annotated data: Universal Dependencies treebanks
- http://universaldependencies.org/
- Earlier: Marcus et al. 1993, The Penn Treebank, *Computational Linguistics*

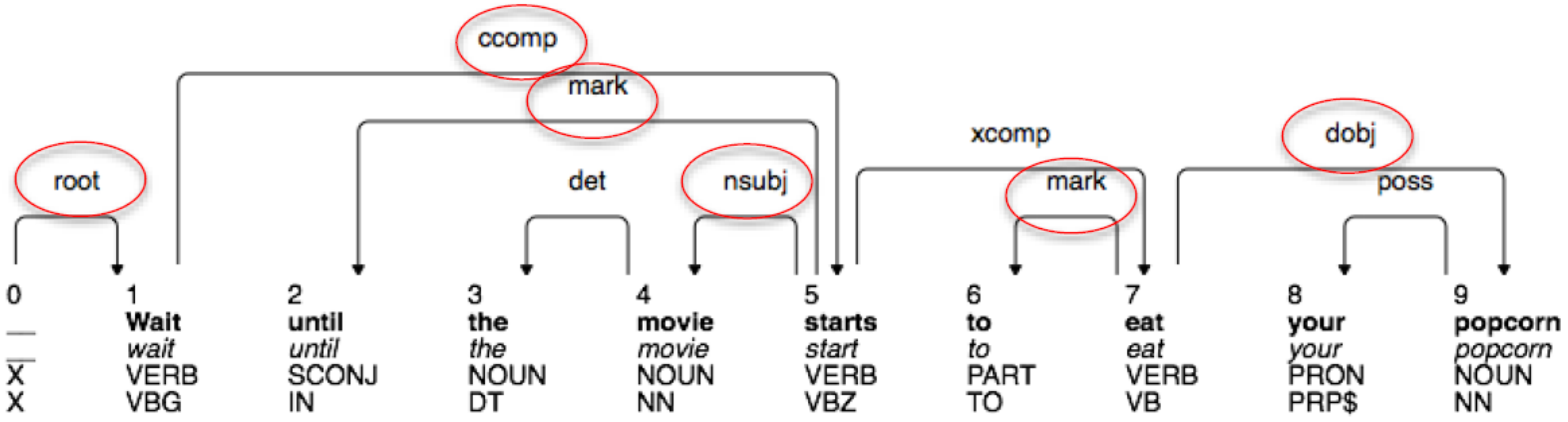


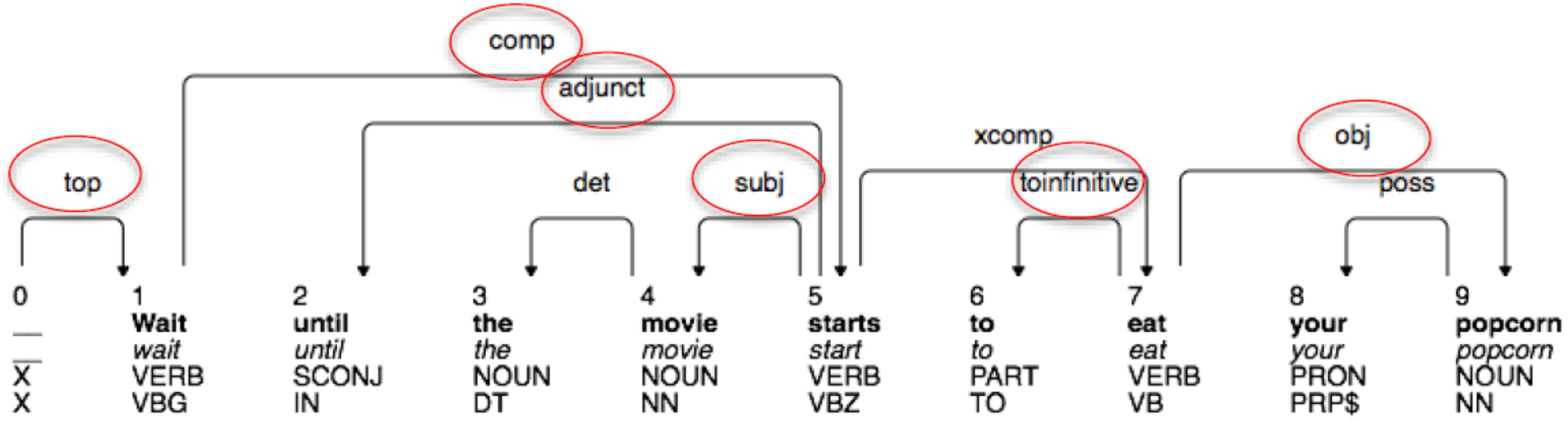
Treebank

- Collection of parsed sentences (trees)
- Annotated with a pre-defined part-of-speech tagset (Noun, Verb, etc.)
- Pre-defined annotation scheme (list of prescribed labels)
- Pre-defined linguistic structure
- Used to develop statistical parsers (train, test, and bootstrap)

Variation in labelling

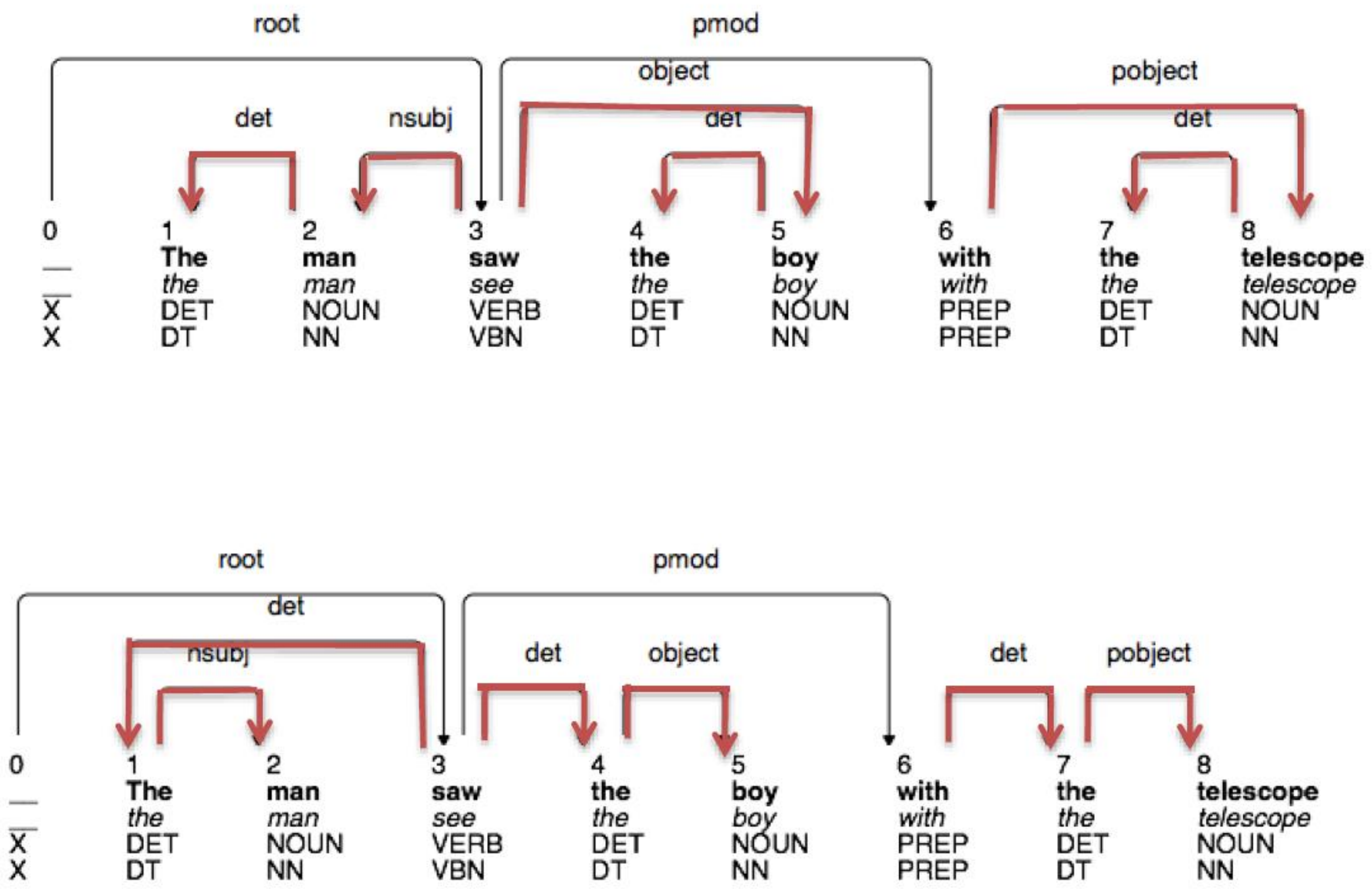
Varying labelling conventions:





Variation in structure

Varying structural analyses:

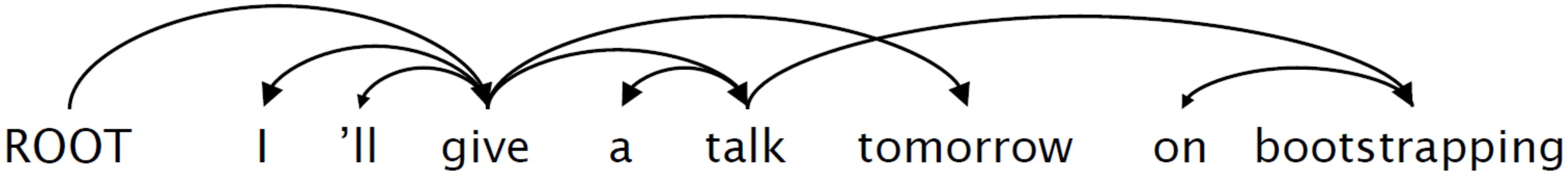


Building treebank

- Building a treebank seems a lot slower and less useful than building a grammar
- But a treebank gives us many things
  - Reusability of the labor
    - Many parsers, part-of-speech taggers, etc. can be built on it
    - Valuable resource for linguistics
  - Broad coverage, not just a few intuitions
  - Frequencies and distributional information
  - A way to evaluate systems

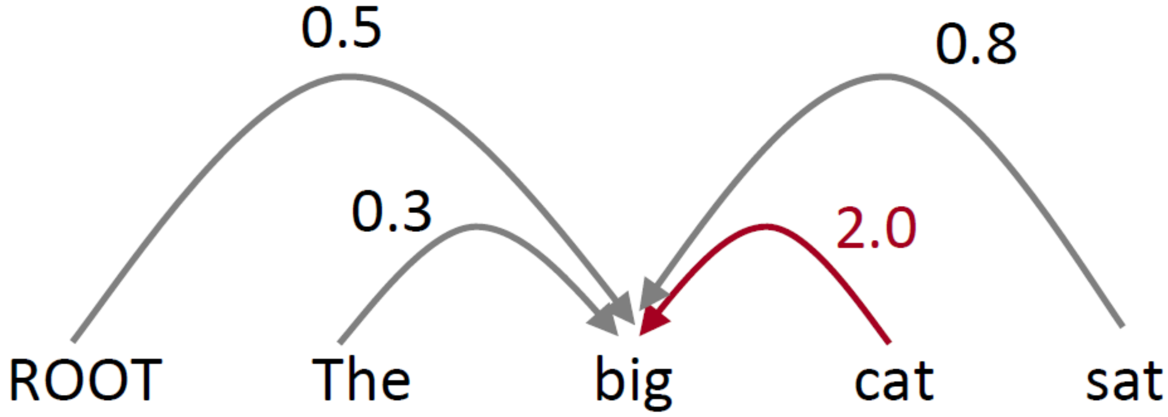
Dependency parsing

- A sentence is parsed by choosing for each word what other word (including ROOT) is it a dependent of
- Usually some constraints:
  - Only one word is a dependent of ROOT
  - Don’t want cycles A → B, B → A
  - This makes the dependencies a tree
  - Final issue is whether arrows can cross (non-projective) or not



Graph-based dependency parsers

- Compute a score for every possible dependency for each word
- Then add an edge from each word to its highest-scoring candidate head
- And repeat the same process for each other word
- E.g., picking the head for “big”



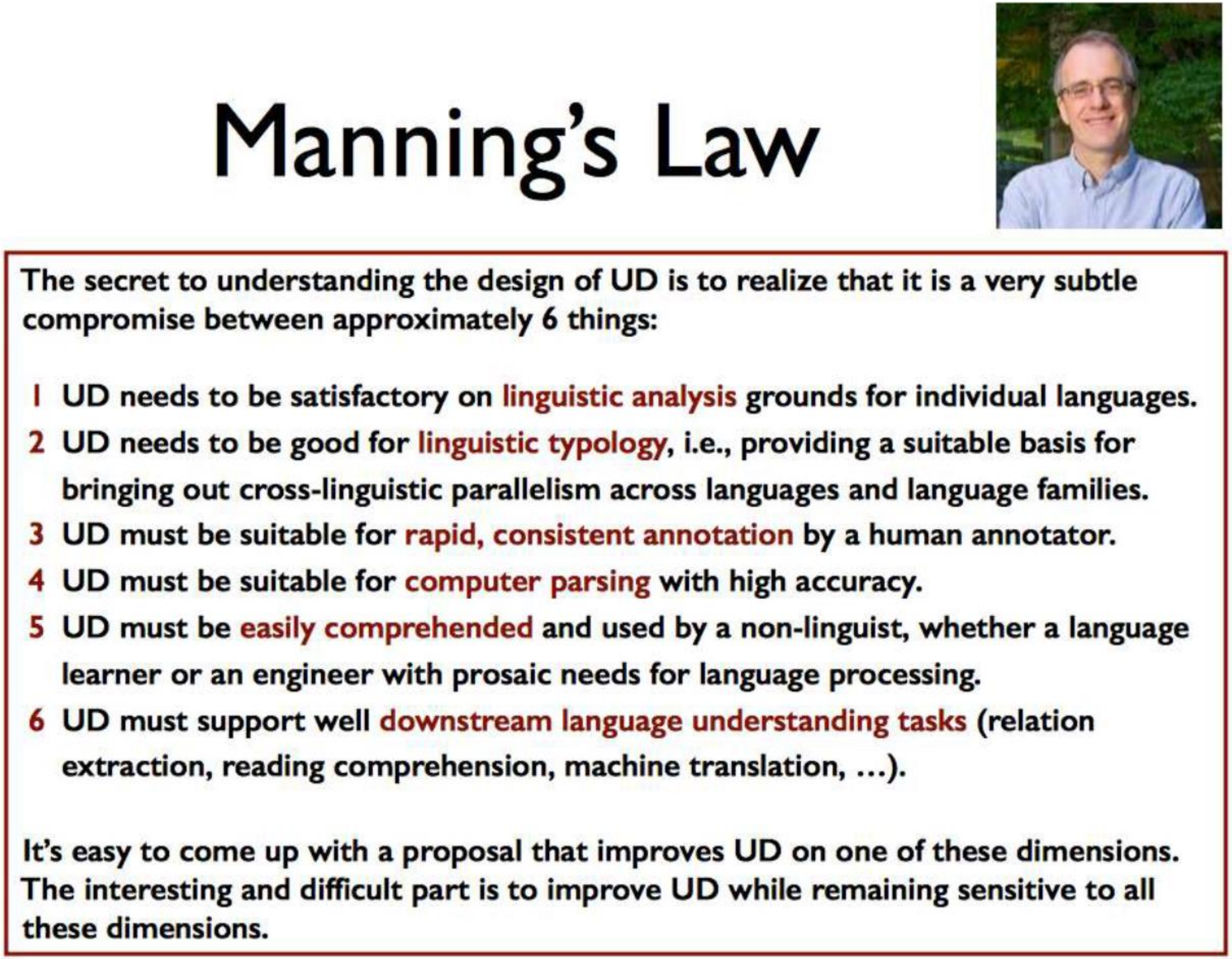
Variation between languages

- **Problems** with variations
- Difficult to do cross-lingual analysis
- Difficult to compare parser performance
- Difficult to do cross-lingual transfer (using data from one language to help another)
- Difficult to build and evaluate multilingual systems

Solution: Universal Dependencies

- ***Premise:***
  - no Universal Grammar, but:
  - “all languages share fundamental similarities” (linguistic universals)
- ***Goals:***
  - develop a set of harmonized dependency treebanks
  - design a universal annotation scheme
  - enable comparison of treebanks
  - enable comparison of parsing results
  - improve multilingual processing

UD creation



Manning's Law

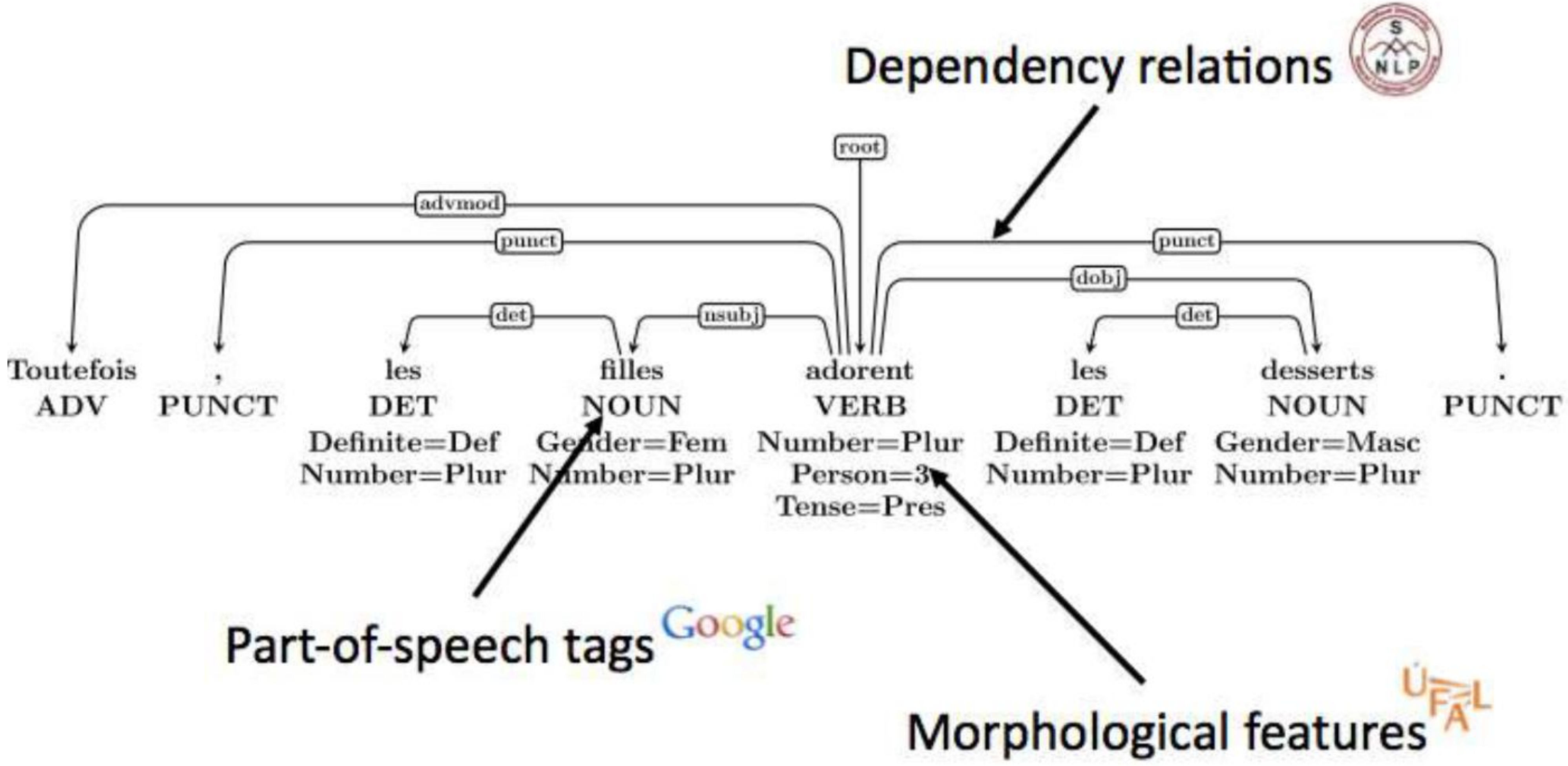
The secret to understanding the design of UD is to realize that it is a very subtle compromise between approximately 6 things:

1. UD needs to be satisfactory on **linguistic analysis** grounds for individual languages.
2. UD needs to be good for **linguistic typology**, i.e., providing a suitable basis for bringing out cross-linguistic parallelism across languages and language families.
3. UD must be suitable for **rapid, consistent annotation** by a human annotator.
4. UD must be suitable for **computer parsing** with high accuracy.
5. UD must be **easily comprehended** and used by a non-linguist, whether a language learner or an engineer with prosaic needs for language processing.
6. UD must support well **downstream language understanding tasks** (relation extraction, reading comprehension, machine translation, …).

It's easy to come up with a proposal that improves UD on one of these dimensions. The interesting and difficult part is to improve UD while remaining sensitive to all these dimensions.

Slide credit: Chris Manning, Stanford University

UD project


Dependency relations
root
advmod
punct
punct
dobj
det
nsubj
det
Toutefois ADV
, PUNCT
les DET Definite=Def Number=Plur
filles NOUN Gender=Fem Number=Plur
adorent VERB Number=Plur Person=3 Tense=Pres
les DET Definite=Def Number=Plur
desserts NOUN Gender=Masc Number=Plur
. PUNCT
Part-of-speech tags
Google
Morphological features
ÚFAL

UD POS tags

- Taxonomy of 17 universal part-of-speech tags, expanding on the Google Universal Tagset (Petrov et al., 2012)
- All languages use the same inventory, but not all tags have to be used by all languages

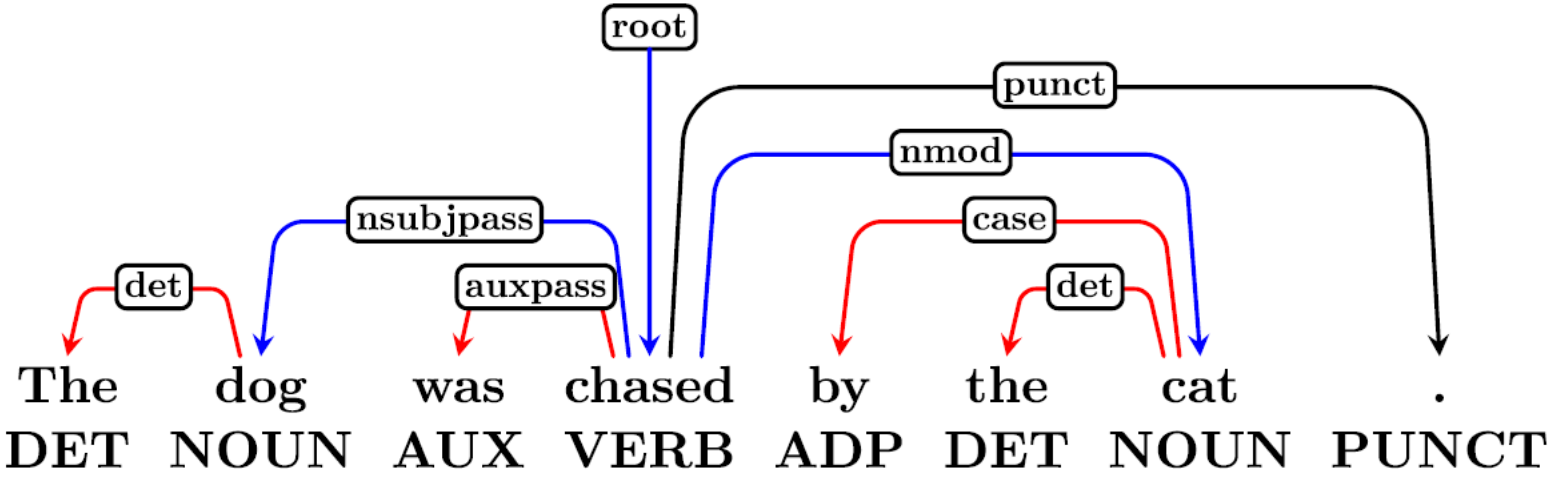
Open	Closed	Other
ADJ	ADP	PUNCT
ADV	AUX	SYM
INTJ	CCONJ	X
NOUN	DET	
PROPN	NUM	
VERB	PART	
	PRON	
	SCONJ	

Slovene UD POS tags

- ADJ: adjective
- ADP: adposition
- ADV: adverb
- AUX: auxiliary verb
- CONJ: coordinating conjunction
- DET: determiner
- INTJ: interjection
- NOUN: noun
- NUM: numeral
- PART: particle
- PRON: pronoun
- PROPN: proper noun
- PUNCT: punctuation
- SCONJ: subordinating conjunction
- SYM: symbol
- VERB: verb
- X: other

UD syntax

- Content words are related by dependency relations
- Function words attach to the content word they further specify
- Punctuation attaches to head of phrase or clause

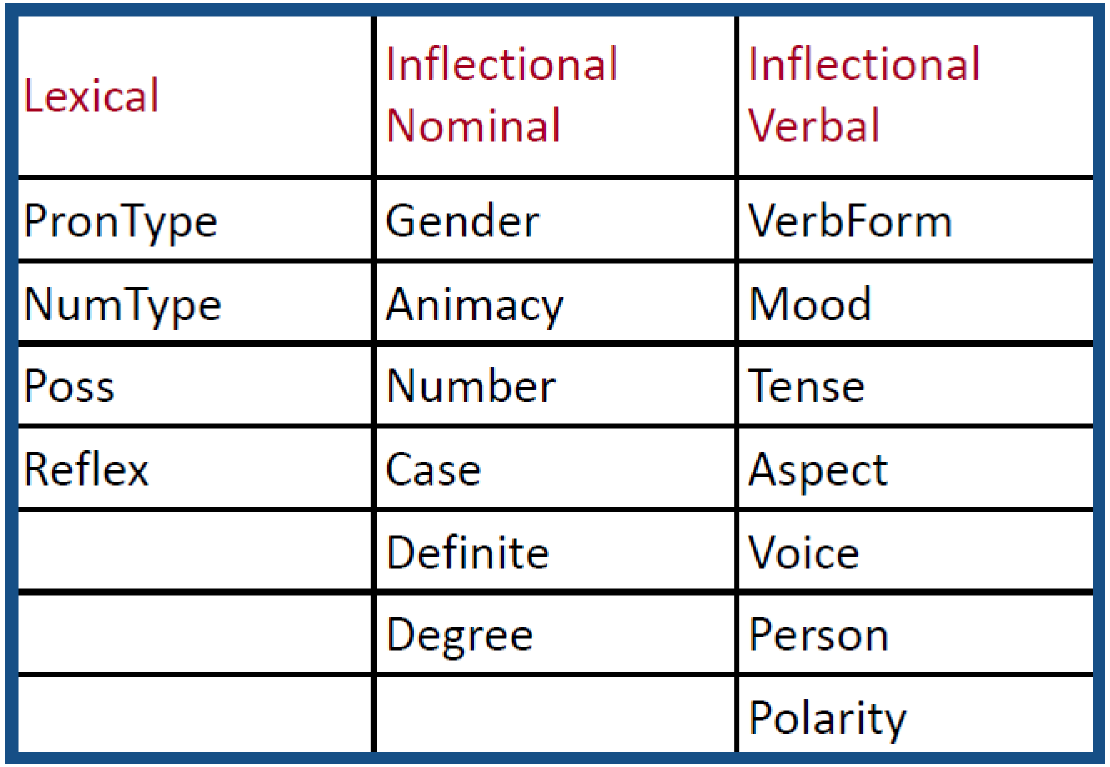


UD relations

- 40 universal grammatical relations (de Marneffe et al., 2014) (aim to address linguistic universals across languages)
- Language-specific subtypes may be added

UD Features

- Standardized inventory of morphological features, based on the Interset system (Zeman, 2008)
- Languages select relevant features and can add language-specific features or values with documentation



Lexical	Inflectional Nominal	Inflectional Verbal
PronType	Gender	VerbForm
NumType	Animacy	Mood
Poss	Number	Tense
Reflex	Case	Aspect
	Definite	Voice
	Degree	Person
		Polarity

Slovene UD features

- **POS Tags**

ADJ – ADP – ADV – AUX – CCONJ – DET – INTJ – NOUN – NUM – PART – PRON – PROPN – PUNCT – SCONJ – VERB – X

- **Features**

Animacy – Aspect – Case – Definite – Degree – Foreign – Gender – Gender[psor] – Mood – Number – Number[psor] – NumForm – NumType – Person – Polarity – Poss – PronType – Tense – Variant – VerbForm

- **Relations**

acl – advcl – advmod – amod – appos – aux – case – cc – cc:preconj – ccomp – conj – conj:extend – cop – csubj – dep – det – discourse – discourse:filler – dislocated – expl – fixed – flat – flat:foreign – flat:name – goeswith – iobj – mark – nmod – nsubj – nummod – obj – obl – orphan – parataxis – parataxis:discourse – parataxis:restart – punct – reparandum – root – vocative – xcomp

- https://universaldependencies.org/treebanks/sl_sst/index.html

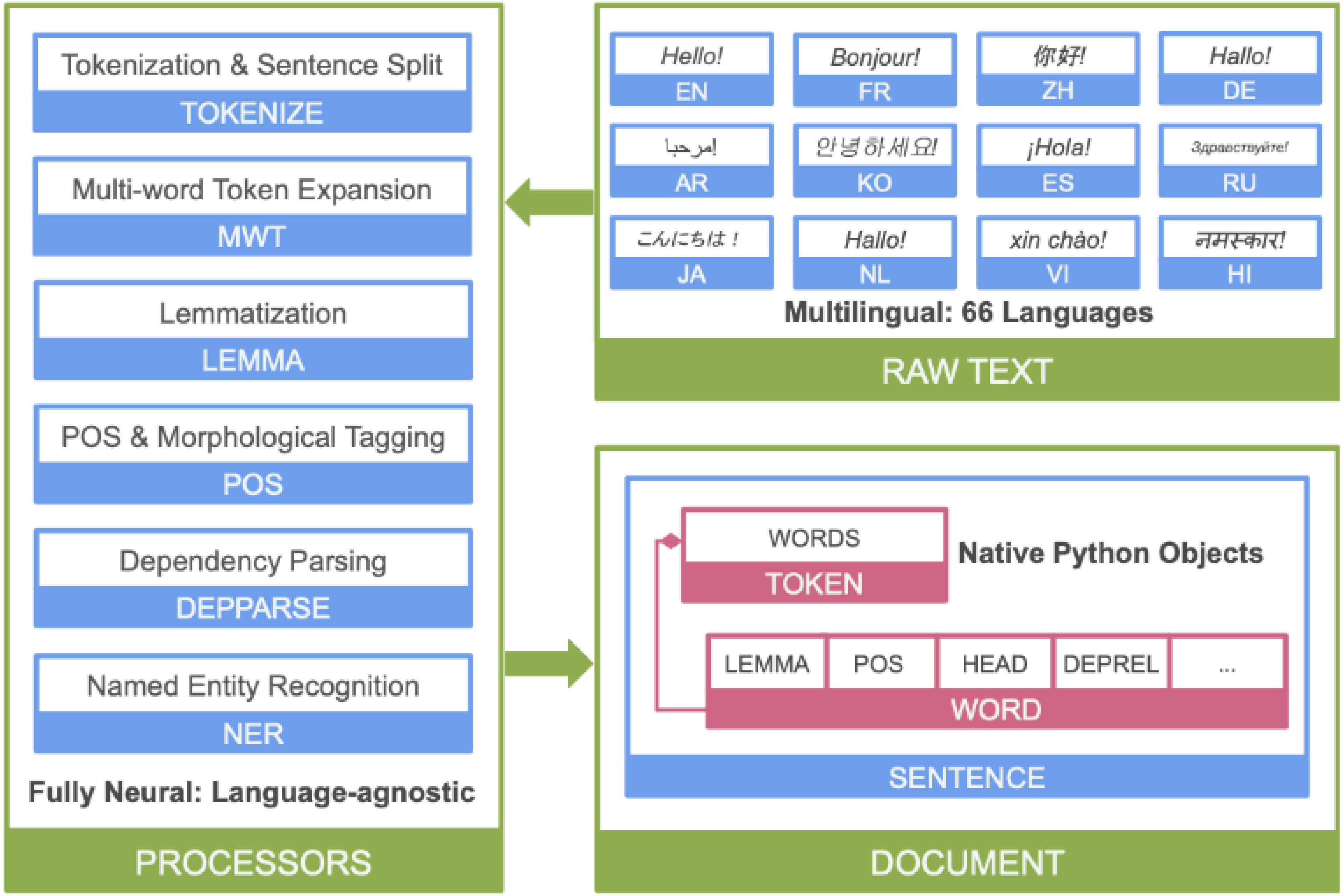
Modern POS and dependency parsing pipelines

- A single neural pipeline for all bottom layer tasks
- Tokenization, sentence and word segmentation, part-of-speech (POS)/morphological features (UFeats)tagging, lemmatization, dependency parsing, and named entity recognition (NER)
- Predominat approach for many languages

Qi, P., Dozat, T., Zhang, Y. and Manning, C.D., 2018, October. Universal Dependency Parsing from Scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (pp. 160-170).

Stanford Stanza pipeline

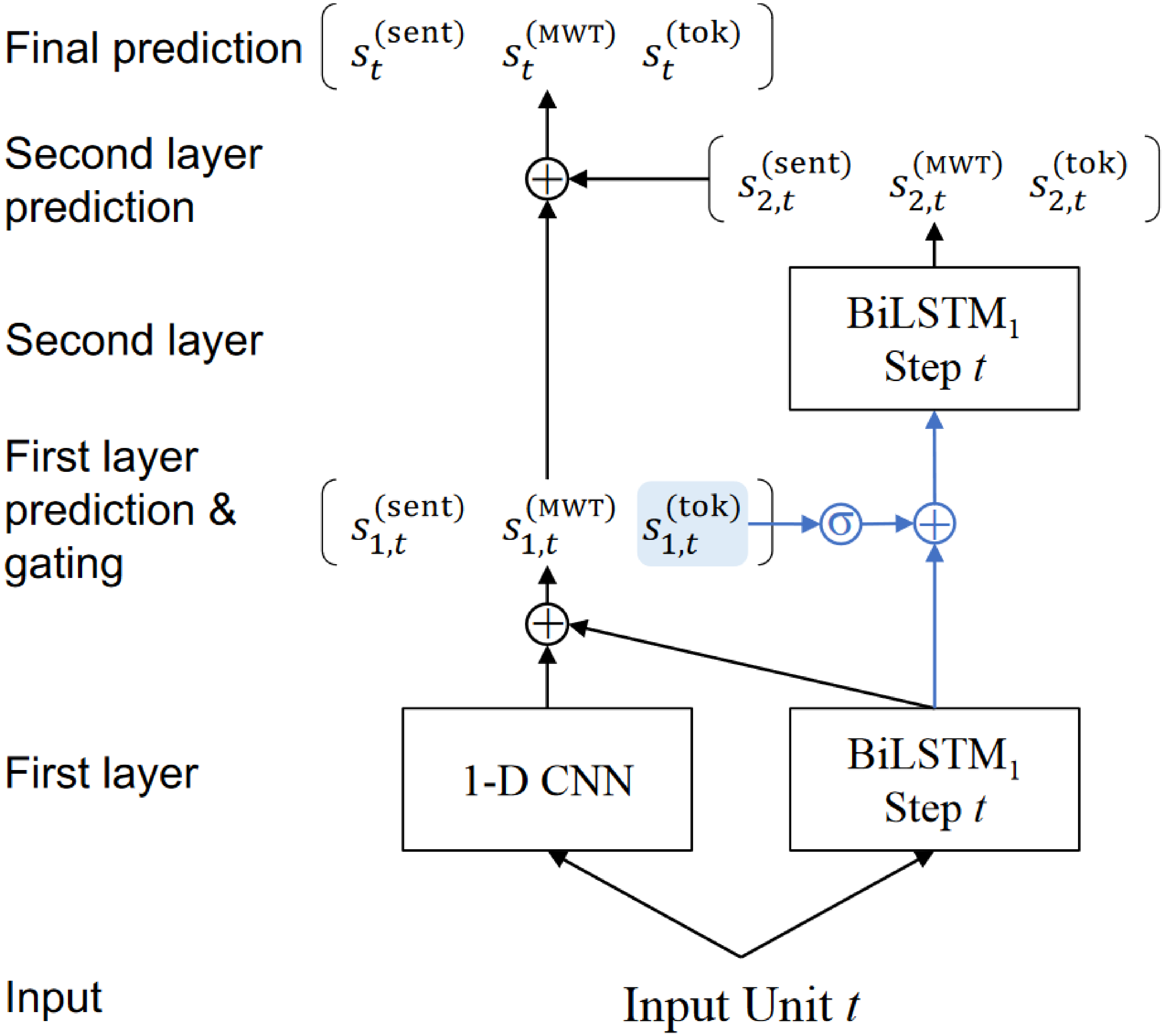
- https://stanfordnlp.github.io/stanza/
- Given a document of raw text,
- The tokenizer/sentence segmenter/MWT expander splits it into sentences of syntactic words;
- The tagger assigns UPOS, XPOS and UFeat tags to each word;
- The lemmatizer takes the predicted word and UPOS tag and outputs a lemma;
- The parser takes all annotations as input and predicts the head and dependency label for each word
- NER is added into the pipeline



Tokenization and sentence segmentation 1/4

- Joint tokenization and sentence segmentation as a unit-level sequence tagging problem
- For most languages, a unit of text is a single character
- Assign one out of five tags to each of the units:
  - end of token (EOT),
  - end of sentence (EOS),
  - multi-word token (MWT),
  - multi-word end of sentence (MWS),and
  - other (OTHER).
- Bidirectional LSTMs(BiLSTMs) as the base model to make unit-level predictions.
- At each unit, the model predicts hierarchically: it first decides whether a given unit is at the end of a token with a score $s^{(tok)}$, then classifies token endings into finer-grained categories with two independent binary classifiers: one for sentence ending $s^{(sent)}$, and one for MWT $s^{(MWT)}$

Tokenization and sentence segmentation 2/4


Final prediction
$s_t^{(\text{sent})}$ $s_t^{(\text{MWT})}$ $s_t^{(\text{tok})}$
Second layer prediction
$s_{2,t}^{(\text{sent})}$ $s_{2,t}^{(\text{MWT})}$ $s_{2,t}^{(\text{tok})}$
Second layer
BiLSTM_1 Step t
First layer prediction & gating
$s_{1,t}^{(\text{sent})}$ $s_{1,t}^{(\text{MWT})}$ $s_{1,t}^{(\text{tok})}$
σ
First layer
1-D CNN
BiLSTM_1 Step t
Input
Input Unit t

Tokenization and sentence segmentation 3/4

- As sentence boundaries and MWTs usually require a larger context, a two-layer BiLSTM is needed
- The first layer BiLSTM operates directly on raw units, and makes the initial prediction over the categories.
- To help capture local unit patterns more easily, the first-layer BiLSTM is combined with 1-D convolutional networks (similar to residual connection) - the output of the CNN is added to the concatenated hidden states of the Bi-LSTM

$$
\begin{aligned}
\mathbf{h}_1^{\mathrm{RNN}} = [\overrightarrow{\mathbf{h}_1}, \overleftarrow{\mathbf{h}_1}] &= \mathrm{BiLSTM}_1(\mathbf{x}), \\
\mathbf{h}_1^{\mathrm{CNN}} &= \mathrm{CNN}(\mathbf{x}), \\
\mathbf{h}_1 &= \mathbf{h}_1^{\mathrm{RNN}} + \mathbf{h}_1^{\mathrm{CNN}}, \\
[\mathbf{s}_1^{(\mathrm{tok})}, \mathbf{s}_1^{(\mathrm{sent})}, \mathbf{s}_1^{(\mathrm{MWT})}] &= W_1 \mathbf{h}_1,
\end{aligned}
$$

Tokenization and sentence segmentation 4/4

- For each unit, concatenate its trainable embedding with a four-dimensional binary feature vector as input:
  1. does the unit start with whitespace;
  2. does it start with a capitalized letter;
  3. is the unit fully capitalized;
  4. is it purely numerical
- To incorporate token-level information at these layer, gating mechanism suppresses representations at non-token boundaries before propagating hidden states upward
- The final prediction concatenates both layers and takes adequate inputs
- Trained with the cross-entropy loss

$$\mathbf{g}_1 = \mathbf{h}_1 \odot \sigma(\mathbf{s}_1^{(\mathrm{tok})})$$

$$\mathbf{h}_2 = [\overrightarrow{\mathbf{h}}_2, \overleftarrow{\mathbf{h}}_2] = \mathrm{BiLSTM}_2(\mathbf{g}_1),$$

$$[\mathbf{s}_2^{(\mathrm{tok})}, \mathbf{s}_2^{(\mathrm{sent})}, \mathbf{s}_2^{(\mathrm{MWT})}] = W_2\mathbf{h}_2,$$

$$p_{\mathrm{EOT}} = p_{+--} \qquad p_{\mathrm{EOS}} = p_{++-}, \tag{8}$$

$$p_{\mathrm{MWT}} = p_{+-+} \qquad p_{\mathrm{MWS}} = p_{+++}, \tag{9}$$

$$\text{where } p_{\pm\pm\pm} = \sigma(\pm s^{(\mathrm{tok})})\sigma(\pm s^{(\mathrm{sent})})\sigma(\pm s^{(\mathrm{MWT})}),$$

$$p_{OTHER} = \sigma(-s^{(tok)})$$

Multi-word Token Expansion

- Tokenizer/sentence segmenter produces a collection of sentences, each being a list of tokens, some of which are labeled as multi-word tokens (MWTs).
- We have to expand these MWTs into the words they correspond to (e.g., “im” to “in dem” in German), in order for downstream systems to process them properly.
- An approach combines symbolic statistical knowledge (lexicon) with the neural system.
- A sequence-to-sequence model using a BiLSTM encoder with an attention mechanism
- The input multi-word token is represented by a sequence of characters $x_1,\ldots,x_I$, and the output syntactic words are represented as a sequence of characters $y_1,\ldots,y_J$, where the words are separated by space characters.
- Inputs to the RNNs are encoded by a shared matrix of character embeddings E.

$$
\begin{aligned}
\mathbf{h}_j^{\text{dec}} &= \text{LSTM}_{\text{dec}}(E_{y_{j-1}}, \mathbf{h}_{j-1}^{\text{dec}}), \\
\alpha_{ij} &\propto \exp(\mathbf{u}_\alpha^\top \tanh(W_\alpha[\mathbf{h}_j^{\text{dec}}, \mathbf{h}_i^{\text{enc}}])), \\
\mathbf{c}_j &= \sum_i \alpha_{ij} \mathbf{h}_i^{\text{enc}}, \\
P(y_j = w | y_{<j}) &\propto \mathbf{u}_w^\top \tanh(W[\mathbf{h}_j^{\text{dec}}, \mathbf{c}_j]).
\end{aligned}
$$

POS/UFeats Tagger

- Highway BiLSTM with inputs coming from the concatenation of three sources:
  1. A pretrained word embedding: word2vec or fastText
  2. A trainable frequent word embedding, for all words that occurred at least seven times in the training set;
  3. A character-level embedding, generated from a unidirectional LSTM over characters in each word.
- UPOS is predicted by first transforming each word’s BiLSTM state with a fully-connected (FC) layer, then applying softmax
- sSimilarly for language specific XPOS, but to ensure consistency between UPOS and XPOS tag sets (e.g., to avoid a VERB UPOS with an NN XPOS), adds UPOS embedding
- Similarly for UFeats with separate parameters for each individual UFeat tag.

$$\mathbf{h}_i = \mathrm{BiLSTM}_i^{(\mathrm{tag})}(\mathbf{x}_1, \ldots, \mathbf{x}_n),$$

$$\mathbf{v}_i^{(\mathrm{u})} = \mathrm{FC}^{(\mathrm{u})}(\mathbf{h}_i),$$

$$P\big(y_{ik}^{(\mathrm{u})} \,|\, X\big) = \mathrm{softmax}_k\big(W^{(\mathrm{u})}\mathbf{v}_i^{(\mathrm{u})}\big).$$

$$\mathbf{v}_i^{(\mathrm{x})} = \mathrm{FC}^{(\mathrm{x})}(\mathbf{h}_i),$$

$$\mathbf{s}_i^{(\mathrm{x})} = [E^{(\mathrm{u})}_{y_{i*}^{(\mathrm{u})}}, 1]^\top \mathbf{U}^{(\mathrm{x})}[\mathbf{v}_i^{(\mathrm{x})}, 1],$$

$$P\big(y_{ik}^{(\mathrm{x})} \,|\, y_{i*}^{(\mathrm{u})}, X\big) = \mathrm{softmax}_k\big(\mathbf{s}_i^{(\mathrm{x})}\big).$$

Lemmatizer 1/2

- Builds two dictionaries from the training set,
  - 1) from a (word, UPOS) pair to the lemma,
  - 2) from the word itself to the lemma.
- During evaluation, the predicted UPOS is used; when the UPOS-augmented dictionary fails, we fall back to the word-only dictionary before resorting to the neural system.
- In looking up both dictionaries, the word is not lower-cased, because case information is more relevant in lemmatization than in MWT expansion
- The neural system is enhanced with an edit classifier that shortcuts the prediction process to accommodate rare, long words, on which the decoder is more likely to flounder.

Lemmatizer 2/2

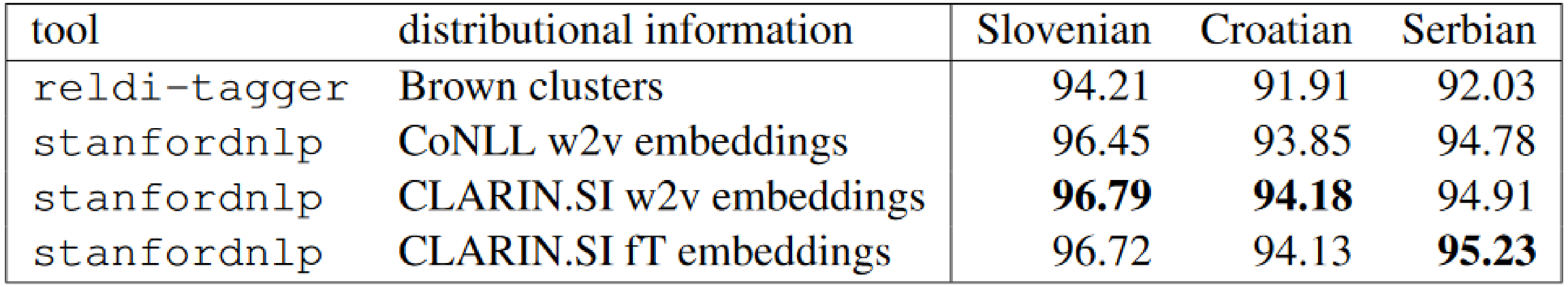
- The concatenated encoder final states are put through an FC layer with ReLU nonlinearity and fed into a 3-way classifier, which predicts whether the lemma is
  1. exactly identical to the word (e.g., URLs and emails),
  2. the lowercased version of the word (e.g., capitalized rare words in English that are not proper nouns), or
  3. in need of the sequence-to-sequence model to make more complex edits to the character sequence.
- During training, we assign the labels to each word-lemma pair greedily in the order of identical, lowercase, and sequence decoder, and train the classifier jointly with the sequence-to-sequence lemmatizer.
- At evaluation time, predictions are made sequentially, i.e., the classifier first determines whether any shortcut can be taken, before the sequence decoder model is used if needed.

Dependency parser

- The high-way BiLSTM takes as input pretrained word embeddings, frequent word and lemma embeddings, character-level word embeddings, summed XPOS and UPOS embeddings, and summed UFeats embeddings.
- First unlabeled dependencies are predicted by scoring each word *i* and its potential heads

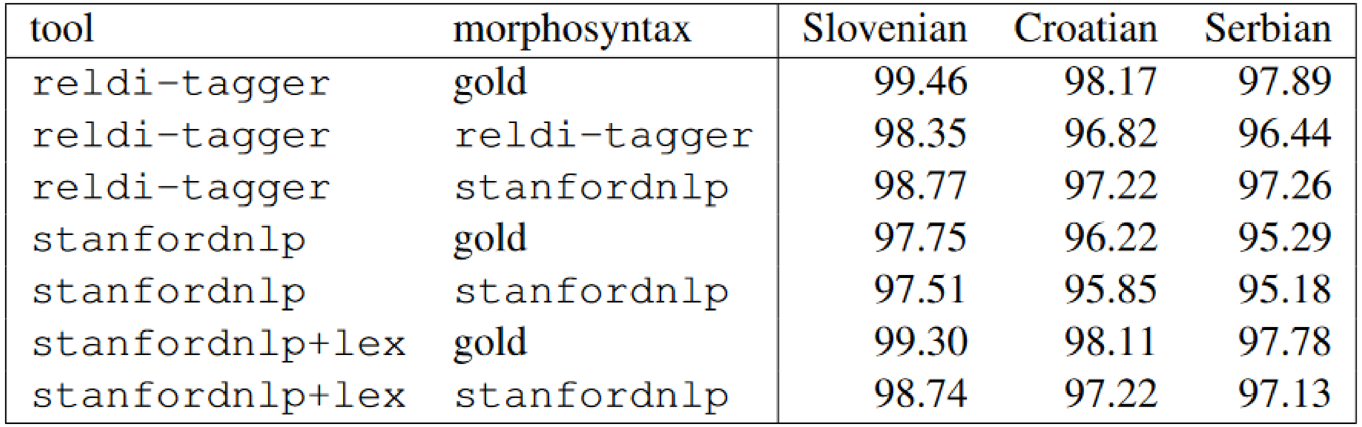
$$\begin{aligned}
\mathbf{h}_t &= \mathrm{BiLSTM}_t^{(\mathrm{parse})}(\mathbf{x}_1, \ldots, \mathbf{x}_n), \\
\mathbf{v}_i^{(\mathrm{ed})}, \mathbf{v}_j^{(\mathrm{eh})} &= \mathrm{FC}^{(\mathrm{ed})}(\mathbf{h}_i), \mathrm{FC}^{(\mathrm{eh})}(\mathbf{h}_j), \\
s_{ij}^{(\mathrm{e})} &= [\mathbf{v}_j^{(\mathrm{eh})}, 1]^\top U^{(\mathrm{e})} [\mathbf{v}_i^{(\mathrm{ed})}, 1], \\
&= \mathrm{Deep\text{-}Biaff}^{(\mathrm{e})}(\mathbf{h}_i, \mathbf{h}_j), \\
P(y_{ij}^{(\mathrm{e})} | X) &= \mathrm{softmax}_j(\mathbf{s}_i^{(\mathrm{e})}),
\end{aligned}$$

Quality of tools in Slovene



tool	distributional information	Slovenian	Croatian	Serbian
reldi-tagger	Brown clusters	94.21	91.91	92.03
stanfordnlp	CoNLL w2v embeddings	96.45	93.85	94.78
stanfordnlp	CLARIN.SI w2v embeddings	**96.79**	**94.18**	94.91
stanfordnlp	CLARIN.SI fT embeddings	96.72	94.13	**95.23**

Table 1: F1 results in morphosyntactic annotation with the traditional and neural tool and different distributional information.



tool	morphosyntax	Slovenian	Croatian	Serbian
reldi-tagger	gold	99.46	98.17	97.89
reldi-tagger	reldi-tagger	98.35	96.82	96.44
reldi-tagger	stanfordnlp	98.77	97.22	97.26
stanfordnlp	gold	97.75	96.22	95.29
stanfordnlp	stanfordnlp	97.51	95.85	95.18
stanfordnlp+lex	gold	99.30	98.11	97.78
stanfordnlp+lex	stanfordnlp	98.74	97.22	97.13

Table 3: F1 results in lemmatisation with the traditional and neural tool and different upstream processing.

Ljubešić, N. and Dobrovoljc, K., 2019. What does Neural Bring? Analysing Improvements in Morphosyntactic Annotation and Lemmatisation of Slovenian, Croatian and Serbian. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing* (pp. 29-34).

Slovene Classla (Stanza) pipeline results 10 January 2022

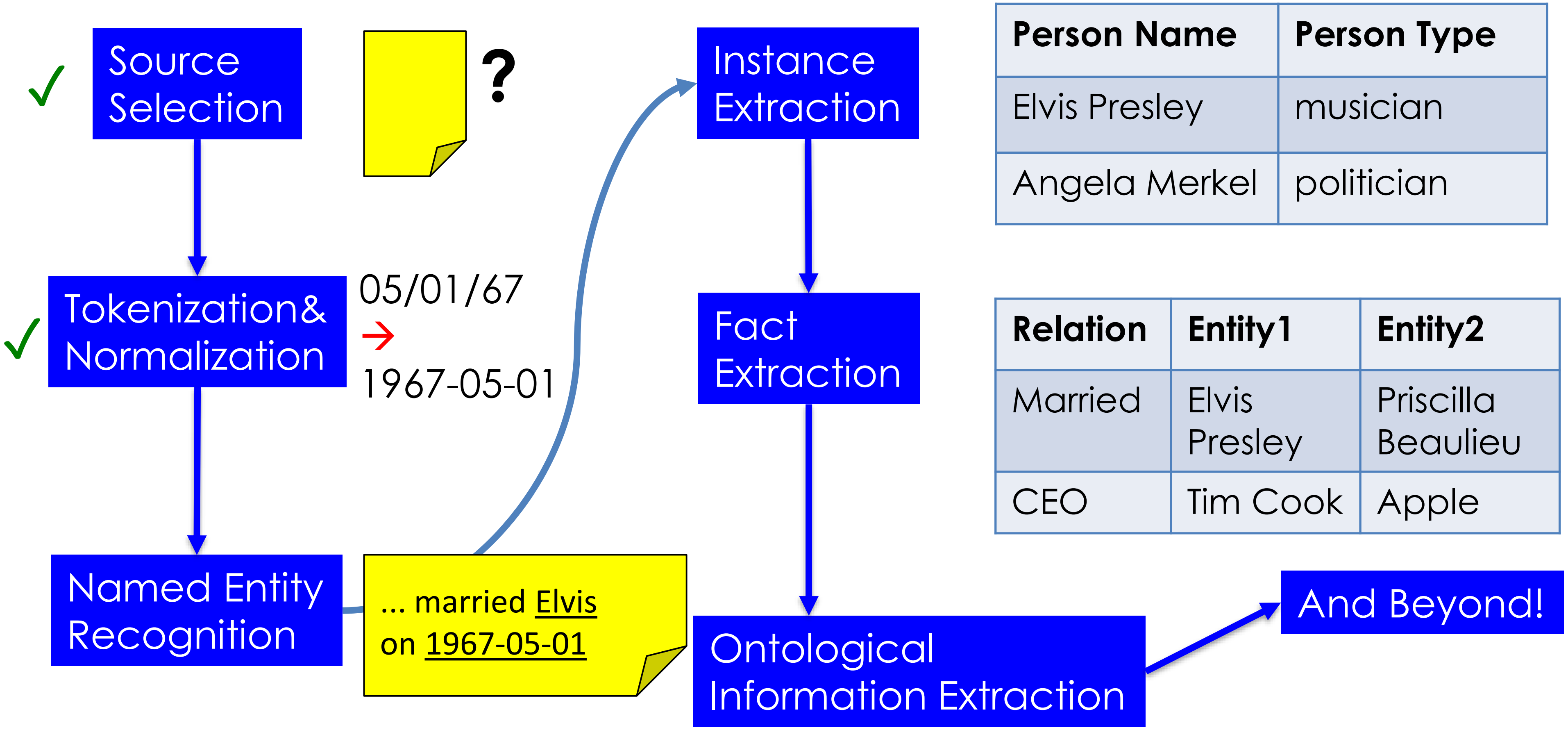
METRIC	PRECISION	RECALL	F1 SCORE	ALIGNDACC
TOKENS	99.97	99.95	99.96	
SENTENCES	99.58	99.47	99.52	
WORDS	99.97	99.95	99.96	
UPOS	98.70	98.69	98.69	98.73
XPOS	97.39	97.37	97.38	97.42
UFEATS	97.01	96.99	97.00	97.04
ALLTAGS	96.33	96.31	96.32	96.36
LEMMAS	99.17	99.16	99.17	99.20
UAS	94.06	94.04	94.05	94.08
LAS	92.05	92.04	92.05	92.08
CLAS	89.34	90.04	89.69	90.09
MLAS	85.08	85.76	85.42	85.80
BLEX	88.75	89.45	89.10	89.50

Named entity recognition (NER)

- Recently, NER was added to the the basic linguistic annotation pipeline
- Why?

Information Extraction

Information Extraction (IE) is the process of extracting **structured information** from unstructured machine-readable documents

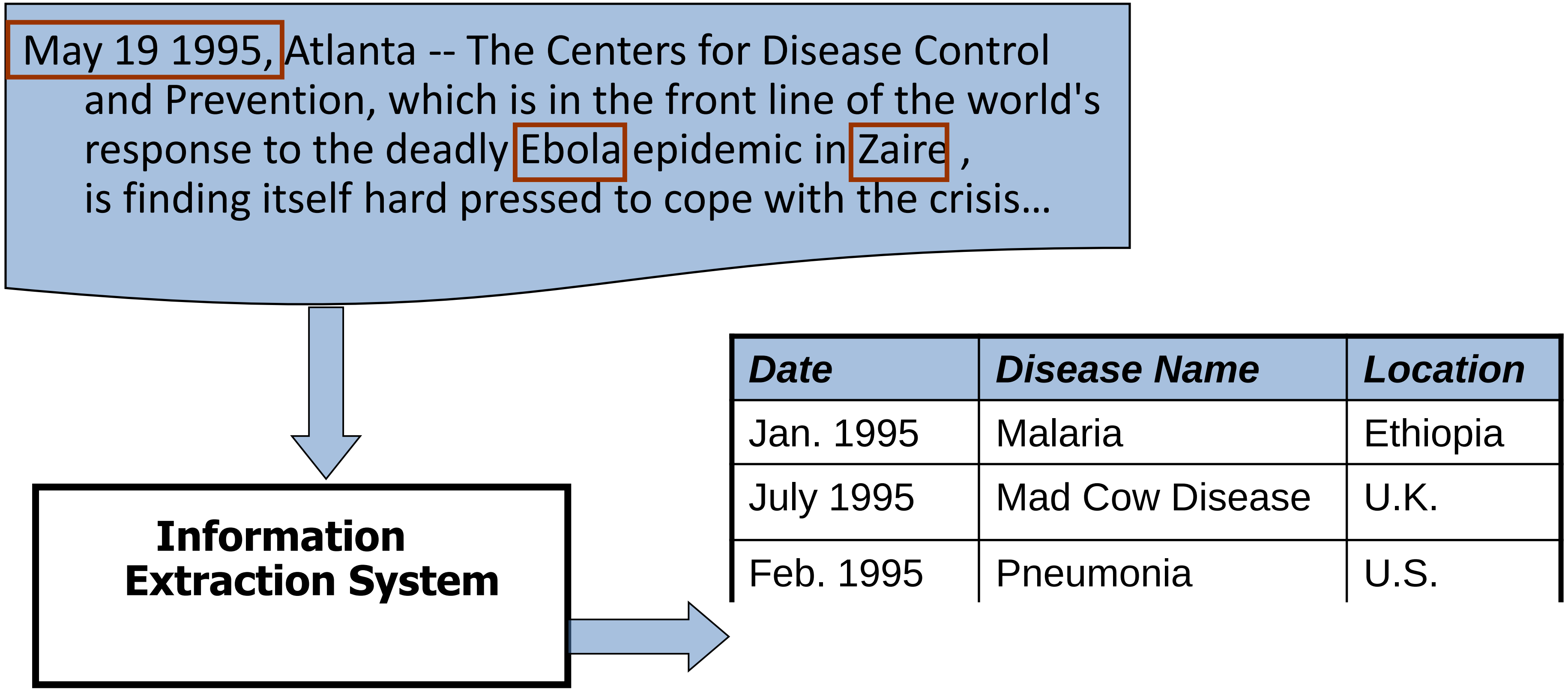


... married Elvis on 1967-05-01

Person Name	Person Type
Elvis Presley	musician
Angela Merkel	politician

Relation	Entity1	Entity2
Married	Elvis Presley	Priscilla Beaulieu
CEO	Tim Cook	Apple

Relation Extraction: Disease Outbreaks



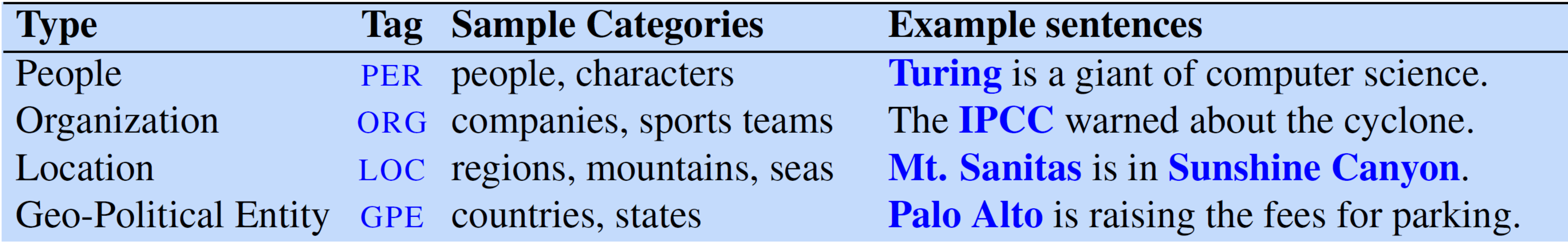
May 19 1995, Atlanta -- The Centers for Disease Control and Prevention, which is in the front line of the world's response to the deadly Ebola epidemic in Zaire , is finding itself hard pressed to cope with the crisis…

Date	***Disease Name***	***Location***
Jan. 1995	Malaria	Ethiopia
July 1995	Mad Cow Disease	U.K.
Feb. 1995	Pneumonia	U.S.

Slide from Manning

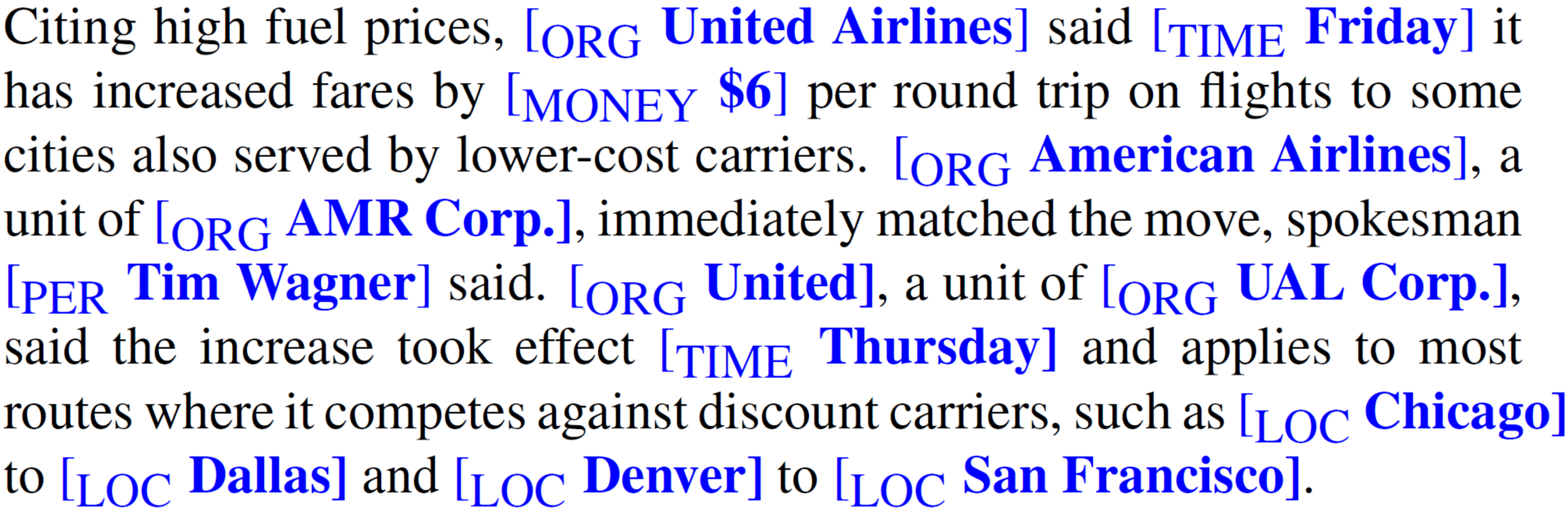
Named entity recognition

- A **named entity** is anything that can be referred to with a **proper name**:
  - a person, a location, an organization.
- **Named entity recognition** (NER) aims to find spans of text that constitute proper names and tag the type of NER entity.
- Four common entity tags:
  - **PER** (person), **LOC** (location), **ORG** (organization), or **GPE** (geo-political entity), **OTHER** (everything else)
- Commonly extended to dates, times, other temporal expressions, numerical expressions like prices.
- Also events, movie and book names, etc.



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.

NER output



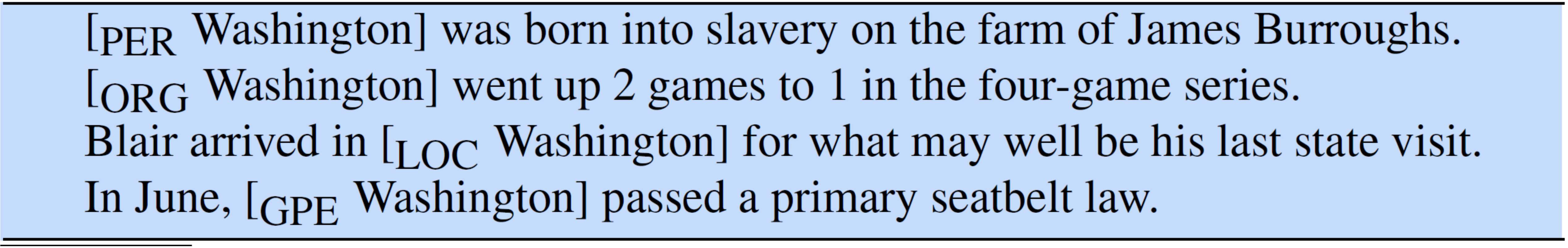
Citing high fuel prices, [$_{\text{ORG}}$ **United Airlines**] said [$_{\text{TIME}}$ **Friday**] it has increased fares by [$_{\text{MONEY}}$ **$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{\text{ORG}}$ **American Airlines**], a unit of [$_{\text{ORG}}$ **AMR 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**].

NER usefulness

- A useful first stage in question answering,
- Linking text to information in structured knowledge sources like Wikipedia.
- Natural language understanding
- Building semantic representations, like extracting events and the relationship between participants.

NER problems

- Ambiguity

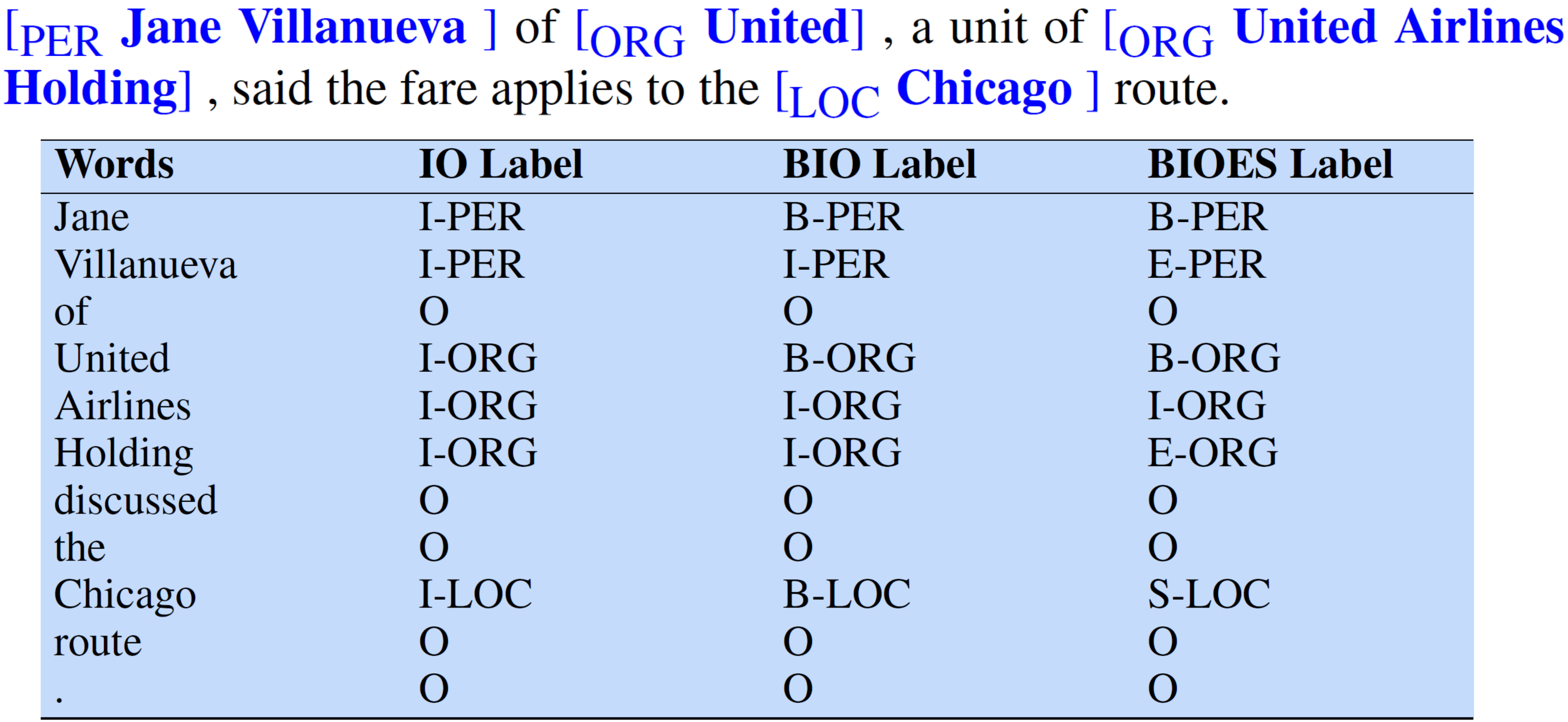


[$_{\text{PER}}$ Washington] was born into slavery on the farm of James Burroughs.
[$_{\text{ORG}}$ Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [$_{\text{LOC}}$ Washington] for what may well be his last state visit.
In June, [$_{\text{GPE}}$ Washington] passed a primary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

- Conceptual dilemmas:
  Republicans were angy because of the reform.
  PER (people of that conviction) or PER (members of that party) or ORG (Republican party) – shall all be labelled at all?
- More complications and ambiguities if PRODUCT is added as a category,
- e,g., Economist (as a physical newspaper or an organization)

NER is a sequence tagging task

- IO, BIO, and BIOES tagging
- for n different tags, the number of labels is: IO=n+1 BIO=2n+1   BIOES:4n+1



[$_{\text{PER}}$ **Jane Villanueva**] of [$_{\text{ORG}}$ **United**] , a unit of [$_{\text{ORG}}$ **United Airlines Holding**] , said the fare applies to the [$_{\text{LOC}}$ **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

NER Evaluation

Comparison to the **gold standard** (i.e. manually labelled or checked output).

Algorithm output:

O = {Einstein, Bohr, Planck, Clinton, Obama}

✓ ✓ ✓ ✗ ✗

Gold standard:

G = {Einstein, Bohr, Planck, Heisenberg}

✓ ✓ ✓ ✗

Precision:

What proportion of the output is correct?

$$\frac{| O \wedge G |}{|O|}$$

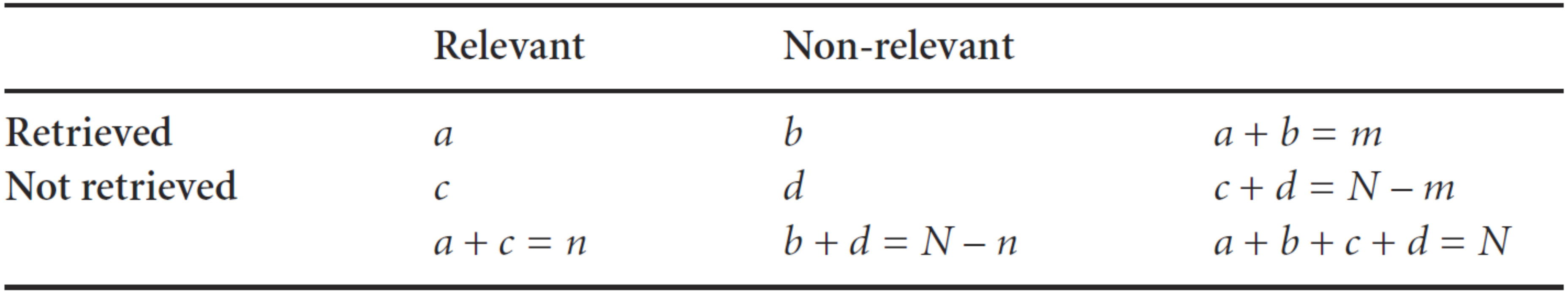
Recall:

What proportion of the gold standard did we get?

$$\frac{| O \wedge G |}{|G|}$$

Performance measures

- A contingency table for the analysis of precision and recall



	Relevant	Non-relevant	
Retrieved	a	b	$a + b = m$
Not retrieved	c	d	$c + d = N - m$
	$a + c = n$	$b + d = N - n$	$a + b + c + d = N$

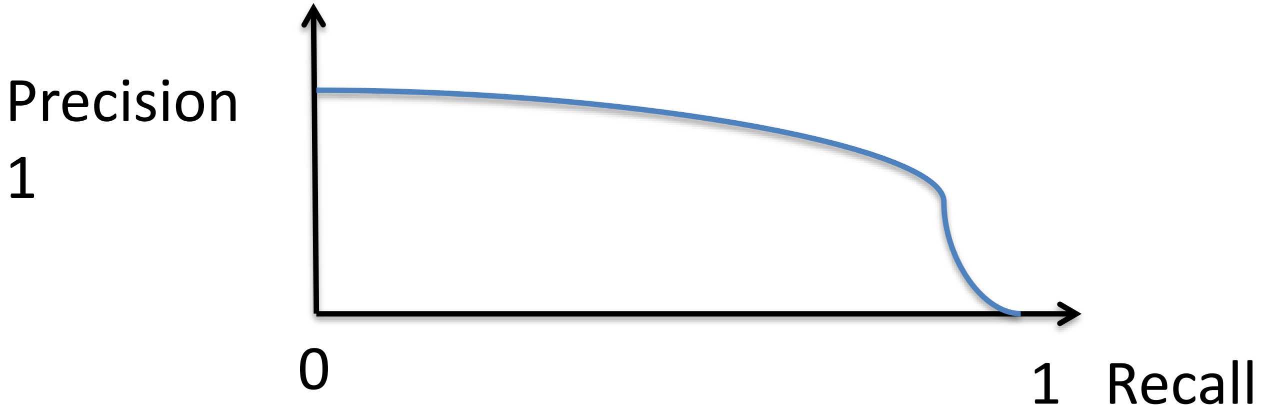
- N = number of all tokens in the dataset
- n = number of relevant tags
- m = number of retrieved tags
- the system returns m tags including a relevant ones
- Precision $P = a/m$

  proportion of relevant tags in the returned ones
- recall $R = a/n$

  proportion of relevant tags in all relevant tags

F1- Measure

You can’t get it all...



The F1-measure combines precision and recall as the harmonic mean:

F1 = 2 * precision * recall / (precision + recall)

NER evaluation dilemmas

- How to treat partial matches?
  - entity may be composed of more than one labelled token
  - training loss (tag based) might not be the same as the test loss (entity based)
- Precision and recall assume two class problems, NER has several tags (at least four)
- The F1 score have to be adapted (micro and macro average variant)
- Micro-average F1: you sum up the individual true positives, false positives, and false negatives of the system for different sets and average them
  - compute several one-versus-all scores and average
  - works well in balanced class case
- Macro-average F1: just take the average of the precision and recall of the system on different set
  - computes TP, FP, TN, FN for each class separately and then compute the measure
  - works better in imbalanced class case
- The *Other* tag is often ignored

Micro and macro averaging example

- Let us compute precision $P = TP / (TP+FP)$.
- Let us assume multi-class classification system with four classes and the following numbers when tested:
- Class A: 1 TP and 1 FP
- Class B: 10 TP and 90 FP
- Class C: 1 TP and 1 FP
- Class D: 1 TP and 1 FP
- $P(A) = P(C) = P(D) = 0.5$, whereas $P(B)=0.1$.
- A macro-averaged precision: $P_{macro} = (0.5+0.1+0.5+0.5) / 4 = 0.4$
- A micro-averaged precision: $P_{micro} = (1+10+1+1) / (2+100+2+2) = 0.123$