Sequence Labeling for Part of Speech and Named Entities

Part of Speech Tagging

Parts of Speech

From the earliest linguistic traditions (Yaska and Panini 5th C. BCE, Aristotle 4th C. BCE), the idea that words can be classified into grammatical categories

- part of speech, word classes, POS, POS tags

8 parts of speech attributed to Dionysius Thrax of Alexandria (c. 1st C. BCE):

- noun, verb, pronoun, preposition, adverb, conjunction, participle, article
- These categories are relevant for NLP today.

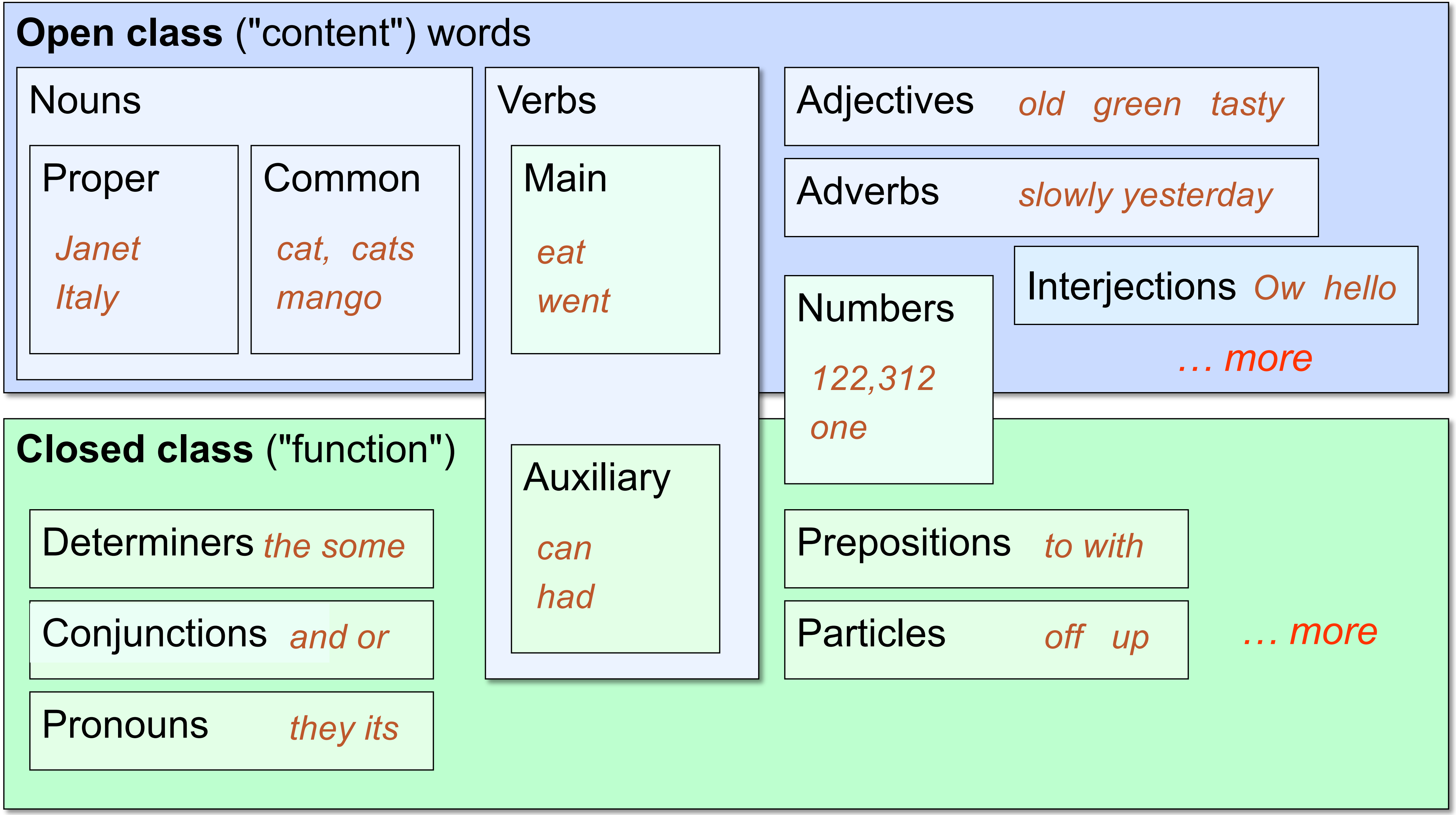
Two classes of words: Open vs. Closed

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
Interjections Ow hello
Numbers
122,312
one
… more
Closed class ("function")
Determiners the some
Conjunctions and or
Pronouns they its
Prepositions to with
Particles off up
… more

Part-of-Speech Tagging

Assigning a part-of-speech to each word in a text.

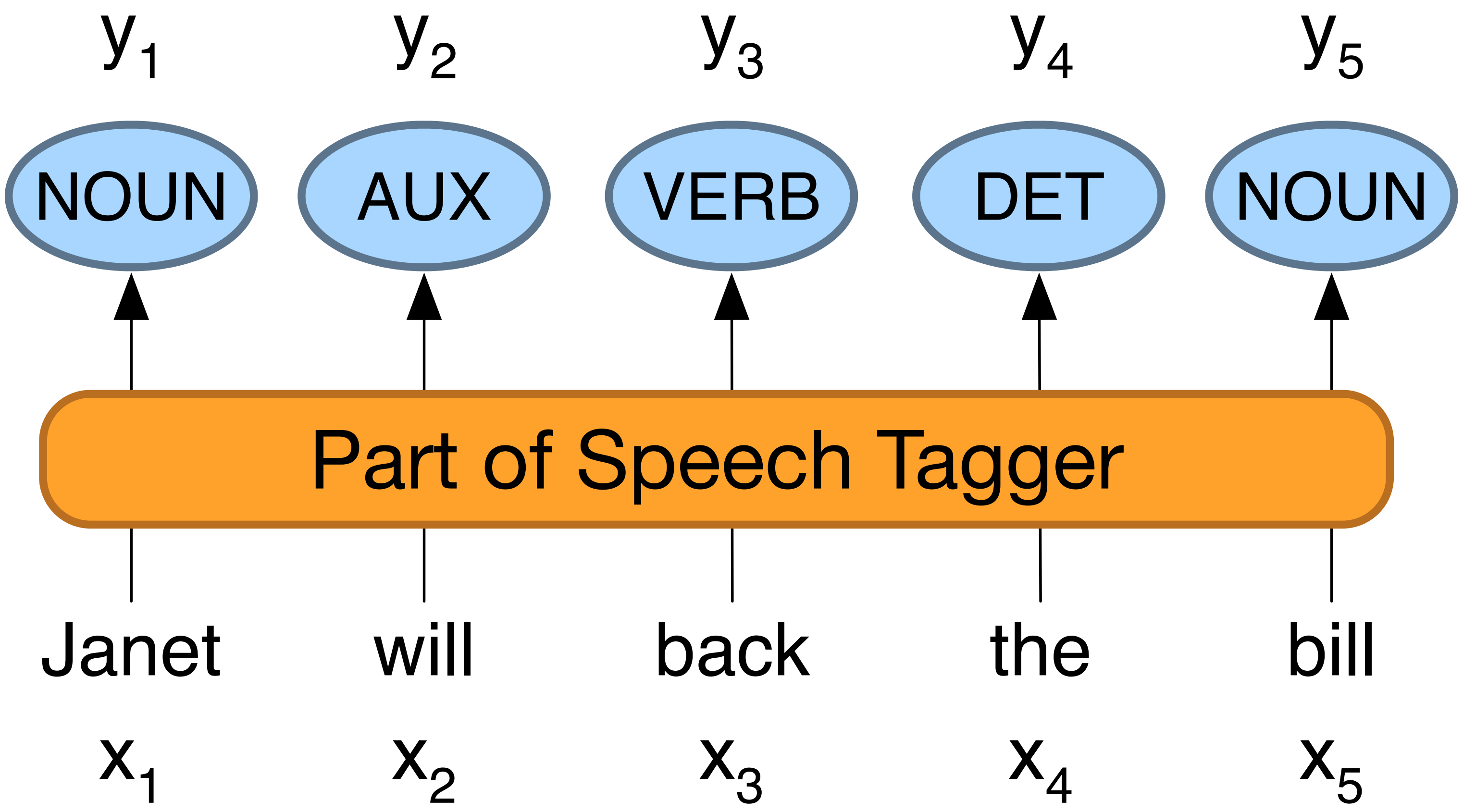
Words often have more than one POS.

book:

- VERB: (***Book** that flight*)
- NOUN: (*Hand me that **book***).

Part-of-Speech Tagging

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



"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

Sample "Tagged" English sentences

There/PRO were/VERB 70/NUM children/NOUN there/ADV ./PUNC

Preliminary/ADJ findings/NOUN were/AUX reported/VERB in/ADP today/NOUN ’s/PART New/PROPN England/PROPN Journal/PROPN of/ADP Medicine/PROPN

Let’s try on line at https://lindat.mff.cuni.cz/services/udpipe/

Sample "Tagged" Italian sentences

Treebank: word used to indicate a corpus where syntactic/semantic sentence structure is annotated so «parsing trees» can be obtained

Assignment:

Parse the following Italian sentences **manually** using the tagset obtained from the UD Italian Stanford Dependency Treebank, and compare with the on line POS tagger:

- Ieri l’altro ho visto Giovanni che prendeva un caffè

- M’illumino d’immenso

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 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

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 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

Sequence Labeling for Part of Speech and Named Entities

Named Entity Recognition (NER)

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

Named Entity tagging

The task of named entity recognition (NER):

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

NER output

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

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

[$_{\text{PER}}$ 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.

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.

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!!!

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

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

Let's take a look to some Italian running models hosted at HuggingFace !!

Standard algorithms for NER/POS tagging

Supervised Machine Learning Algorithms:

- Hidden Markov Models
- *Conditional Random Fields (CRF)*
- 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

HMM

Let's start from the *Markov assumption* for bigrams:

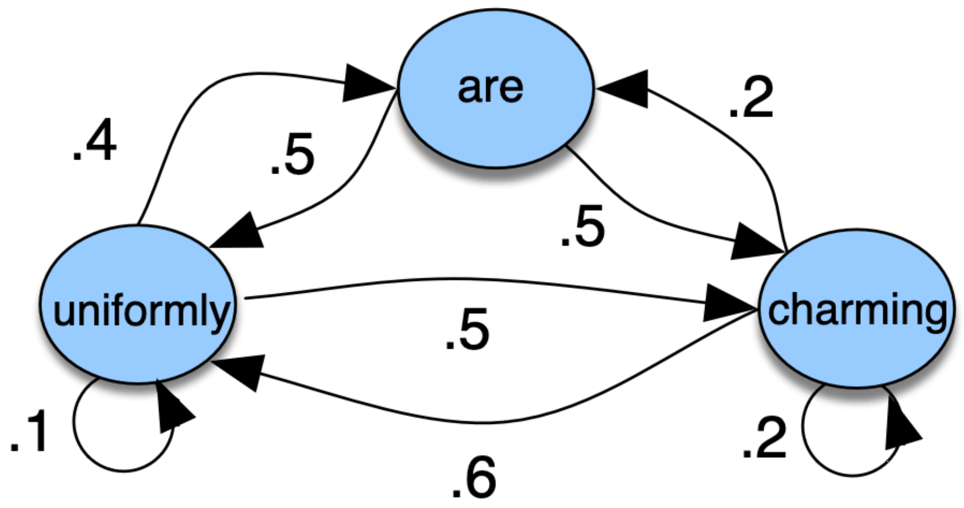
$$P(w_i \mid w_1, w_2, \ldots, w_{i-1}) = P(w_i \mid w_{i-1})$$

It simplifies a model describing the probability for a stochastic system of *being in certain state after having moved through a series of states* that is a

Markov chain

HMM

Markov chain



$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{N1} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	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, $\sum_{i=1}^{n} \pi_i = 1$

HMM

Hidden Markov Model:

A model that uses a Markov Chain to estimate the probability of a series of *hidden events (states)* (i.e. the POS tags to be devised) starting from a series of *observations* (i.e. the words in a sentence) caused from hidden events

HMM

Hidden Markov Model

$Q = q_1 q_2 \dots q_N$	a set of N **states**
$A = a_{11} \dots a_{ij} \dots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \dots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t 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, $\sum_{i=1}^{n} \pi_i = 1$

Markov Assumption: $P(q_i | q_1, ..., q_{i-1}) = P(q_i | q_{i-1})$

→ *First order HMM*

Output Independence: $P(o_i | q_1, \dots q_i, \dots, q_T, o_1, \dots, o_i, \dots, o_T) = P(o_i | q_i)$

HMM Tagger

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

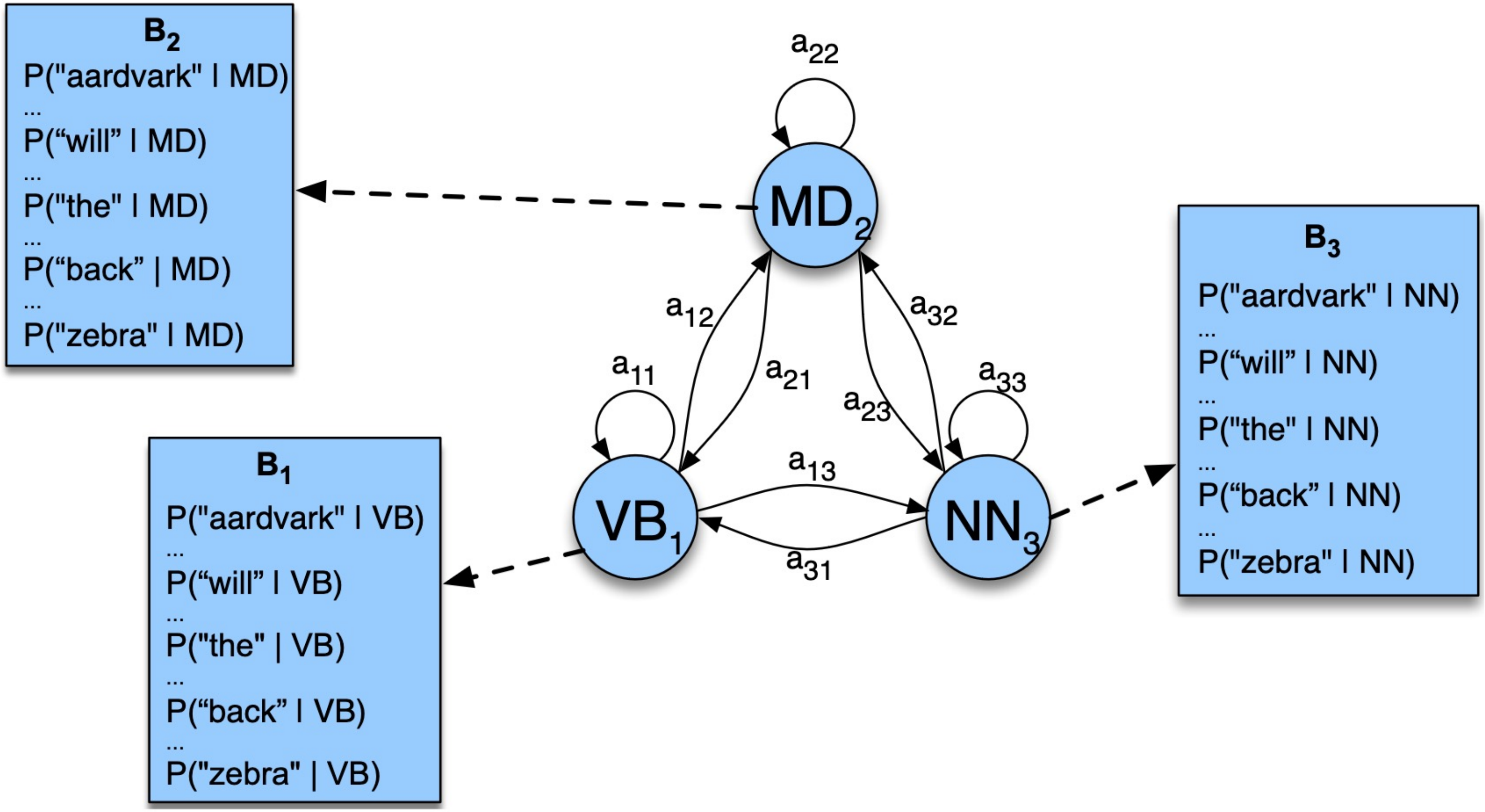
Tag transition probability (A) MLE

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

Tag emission probability (B) MLE

P(*will*|MD): *given that the next tag is MD, how probable is that we observe the word «will»?*

HMM Tagger



B2
P("aardvark" I MD)
...
P("will" I MD)
...
P("the" I MD)
...
P("back" | MD)
...
P("zebra" I MD)
a22
MD2
a12
a21
a32
a23
a11
a33
a13
a31
VB1
NN3
B1
P("aardvark" I VB)
...
P("will" I VB)
...
P("the" | VB)
...
P("back" I VB)
...
P("zebra" | VB)
B3
P("aardvark" I NN)
...
P("will" I NN)
...
P("the" I NN)
...
P("back" I NN)
...
P("zebra" I NN)

HMM Tagger

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n)$$

The problem

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(w_1 \ldots w_n | t_1 \ldots t_n) P(t_1 \ldots t_n)$$

Bayes rule, discarding the evidence $P(w_1, \ldots, w_n)$

$$P(w_1 \ldots w_n | t_1 \ldots t_n) \approx \prod_{i=1}^{n} P(w_i | t_i)$$

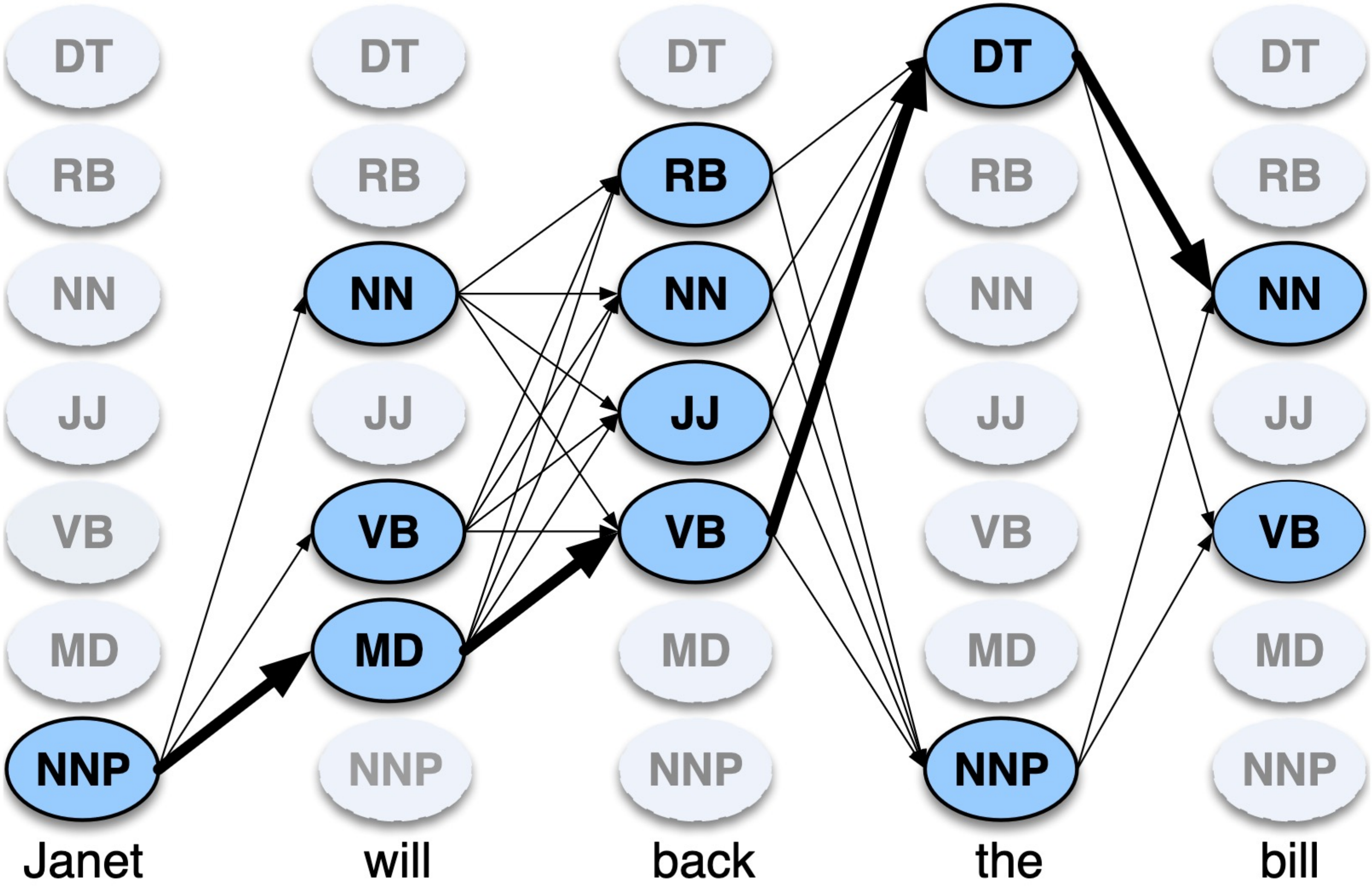
Output independence

$$P(t_1 \ldots t_n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1})$$

Bigram assumption

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n) \approx \underset{t_1 \ldots t_n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i | t_i)}^{\text{emission}} \overbrace{P(t_i | t_{i-1})}^{\text{transition}}$$

HMM Tagger



Viterbi Algorithm: $v_t(j) = \max_{i=1}^{N} v_{t-1}(i) P(t_j \mid t_i) P(w_t \mid t_j) \quad 1 \leq j \leq N, 1 < t \leq T$

CRF

It would be great if we could take into account of arbitrary features in HMM

- Unknown word in POS tagging
- New verbs and (proper/common) nouns
- Morphology rules (i.e. *–ed* → VBD or VBN)
- …

CRF

HMM are not so good in dealing with arbitrary features

→ They are generative models

→ Use pre-computed probabilities: many cond. probabilities to be added for just one new feature

Long feature vectors can be managed better using *discriminative models*

CRF

CRF learns to predict globally the most probable tag sequence $\hat{Y}$ from all possible tag sequences $\mathcal{Y}$ given the sentence X

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(Y|X)$$

It makes use of a multinomial logistic regression (i.e. logistic regression on many classes)

(Linear chain) CRF

Softmax

It depends only on X: does not affect argmax

$$p(Y|X) = \frac{\exp\left(\sum_{k=1}^{K} w_k \boxed{F_k(X,Y)}\right)}{\boxed{\sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y')\right)}} = \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)$$

Global feature

$$F_k(X,Y) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)$$

CRF

<u>*Local features*</u> depend only on the tag couple (y_i, y_{i-1}) the position *i* and the sentence *X*

$$f_k(y_{i-1}, y_i, X, i)$$

$$\mathbb{1}\{x_i = \textit{the},\ y_i = \text{DET}\}$$
$$\mathbb{1}\{y_i = \text{PROPN},\ x_{i+1} = \textit{Street},\ y_{i-1} = \text{NUM}\}$$
$$\mathbb{1}\{y_i = \text{VERB},\ y_{i-1} = \text{AUX}\}$$

1 if the rule holds
0 otherwise

CRF

Feature templates:

- abstract specification of features that can be filled automatically from the corpus

$$\langle y_i, x_i \rangle, \langle y_i, y_{i-1} \rangle, \langle y_i, x_{i-1}, x_{i+2} \rangle$$

Janet/NNP will/MD back/VB the/DT bill/NN

⟨VB, *back*⟩, ⟨VB, MD⟩, ⟨VB, *will*, *bill*⟩

CRF

Word shapes:

- Prefixes, suffixes, multi-word structures

well-dressed

$$\text{prefix}(x_i) = \texttt{w}$$
$$\text{prefix}(x_i) = \texttt{we}$$
$$\text{suffix}(x_i) = \texttt{ed}$$
$$\text{suffix}(x_i) = \texttt{d}$$
$$\text{word-shape}(x_i) = \texttt{xxxx-xxxxxxx}$$
$$\text{short-word-shape}(x_i) = \texttt{x-x}$$

CRF

Typical features for NER:

A list of geographical names

identity of w_i, identity of neighboring words
embeddings for w_i, embeddings for neighboring words
part of speech of w_i, part of speech of neighboring words
presence of w_i in a **gazetteer**
w_i contains a particular prefix (from all prefixes of length ≤ 4)
w_i contains a particular suffix (from all suffixes of length ≤ 4)
word shape of w_i, word shape of neighboring words
short word shape of w_i, short word shape of neighboring words
gazetteer features

CRF

Words	POS	Short shape	Gazetteer	BIO Label
Jane	NNP	Xx	0	B-PER
Villanueva	NNP	Xx	1	I-PER
of	IN	x	0	O
United	NNP	Xx	0	B-ORG
Airlines	NNP	Xx	0	I-ORG
Holding	NNP	Xx	0	I-ORG
discussed	VBD	x	0	O
the	DT	x	0	O
Chicago	NNP	Xx	1	B-LOC
route	NN	x	0	O
.	.	.	0	O

All features can be encoded as binary ones

CRF

Training

- Stochastic Gradient Descent with Cross-Entropy Loss
- Regularization required

Inference

- Viterbi Algorithm where CRF features are *added* to the current Viterbi path

$$v_t(j) \;=\; \max_{i=1}^{N} \; v_{t-1}(i) + \sum_{k=1}^{K} w_k f_k(y_{t-1}, y_t, X, t) \quad 1 \leq j \leq N, 1 < t \leq T$$