Natural Language Processing: Part-of-Speech Tagging, Named Entity Retrieval

21 February 2022

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
  - and also discussion on… syntax, diphthong, clitic, and analogy..
- These categories are relevant for NLP today
- Though actually his 8 aren't exactly the ones we are taught today
  - Thrax: noun, verb, article, adverb, preposition, conjunction, participle, pronoun
  - School grammar: noun, verb, adjective, adverb, preposition, conjunction, pronoun, interjection, etc.

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*
    - Nouns (*Googler, textlish*), Verbs (*Google*), Adjectives (*geeky*), Adverbs (*automagically*)

Open class ("content") words

Nouns

Proper

Janet

Italy

Common

cat cats

mango

beauty

Verbs

Main

eat

went

Auxiliary

can

had

Adjectives *old green tasty*

Adverbs *slowly yesterday here home*

Numbers

122,312

one

Interjections *Ow hello*

… more

Closed class ("function")

Determiners *the some*

Conjunctions *and or that*

Pronouns *they its*

Prepositions *to with*

Particles *off up*

… more

English conjunctions: sampled frequencies from COBUILD

and	514,946	yet	5,040	considering	174	forasmuch as	0
that	134,773	since	4,843	lest	131	however	0
but	96,889	where	3,952	albeit	104	immediately	0
or	76,563	nor	3,078	providing	96	in as far as	0
as	54,608	once	2,826	whereupon	85	in so far as	0
if	53,917	unless	2,205	seeing	63	inasmuch as	0
when	37,975	why	1,333	directly	26	insomuch as	0
because	23,626	now	1,290	ere	12	insomuch that	0
so	12,933	neither	1,120	notwithstanding	3	like	0
before	10,720	whenever	913	according as	0	neither nor	0
though	10,329	whereas	867	as if	0	now that	0
than	9,511	except	864	as long as	0	only	0
while	8,144	till	686	as though	0	provided that	0
after	7,042	provided	594	both and	0	providing that	0
whether	5,978	whilst	351	but that	0	seeing as	0
for	5,935	suppose	281	but then	0	seeing as how	0
although	5,424	cos	188	but then again	0	seeing that	0
until	5,072	supposing	185	either or	0	without	0

CELEX: online dictionary
Frequency counts are from 16-billion-word corpus called COBUILD

English propositions: sampled frequencies from COBUILD

of	540,085	through	14,964	worth	1,563	pace	12
in	331,235	after	13,670	toward	1,390	nigh	9
for	142,421	between	13,275	plus	750	re	4
to	125,691	under	9,525	till	686	mid	3
with	124,965	per	6,515	amongst	525	o'er	2
on	109,129	among	5,090	via	351	but	0
at	100,169	within	5,030	amid	222	ere	0
by	77,794	towards	4,700	underneath	164	less	0
from	74,843	above	3,056	versus	113	midst	0
about	38,428	near	2,026	amidst	67	o'	0
than	20,210	off	1,695	sans	20	thru	0
over	18,071	past	1,575	circa	14	vice	0

CELEX: online dictionary
Frequency counts are from 16-billion-word corpus COBUILD

Penn Treebank part-of-speech 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*

Original Brown corpus used a large set of 87 POS tags.
Most common in NLP today is the Penn Treebank set of 45 tags.
See also info at: https://universaldependencies.org/u/pos/

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn treebank. Computational Linguistics 19(2), 313–330
Nivre, J., et al. (2016a). Universal Dependencies v1: A multilingual tree-bank collection. LREC

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

Common corpora

- Brown Corpus: million words from 500 written texts in 1961
- WSJ: a million words published in the Wall Street Journal in 1989
- Switchboard: 2 million words of telephone conversations 1990-1991

Usually created by running an automatic POS tagger on the texts and correcting each tag manually

Why Part of Speech Tagging?

- Parts of speech are useful clues to sentence structure and meaning
- Knowing whether a word is a noun or a verb tells us about likely neighboring words and syntactic structure, making part-of-speech tagging a key aspect of parsing
  - E.g., nouns in English are preceded by determiners and adjectives, verbs by nouns
  - E.g., verbs have dependency links to nouns

Why Part of Speech Tagging?

- Can be useful for NLP tasks
  - Parsing: POS tagging can improve syntactic parsing, WSD
  - MT: reordering of adjectives and nouns (e.g., from Spanish to English)
  - Sentiment or affective tasks: may want to distinguish adjectives or other POS
  - Word prediction in speech recognition: e.g., possessive pronouns (my, your, her) are likely to be followed by nouns and personal pronouns (I, you, he) are likely to be followed by verbs
  - Text-to-speech (determine how to pronounce “lead” or "object"?)
- Or for 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

Although many neural models don’t use POS tagging, it is still important to understand what makes POS tagging difficult (or easy), and how the basic models and algorithms work

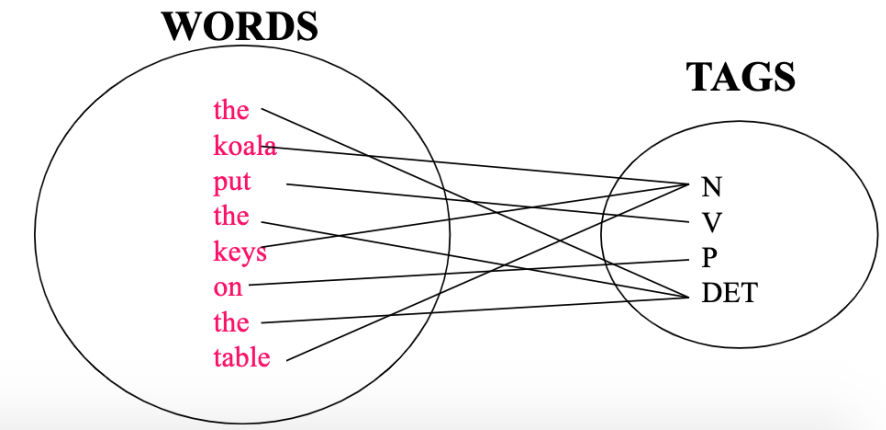
POS is indicative of pronunciation

Noun	Verb
My conduct is great	I conduct myself well
She won the contest	I contest the ticket
He is my escort	He escorted me
That is an insult	Don’t insult me
Rebel without a cause	He likes to rebel
He is a suspect	I suspect him

Part-of-Speech Tagging

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

  The POS tagging problem is to determine the POS tag for a particular instance of a word

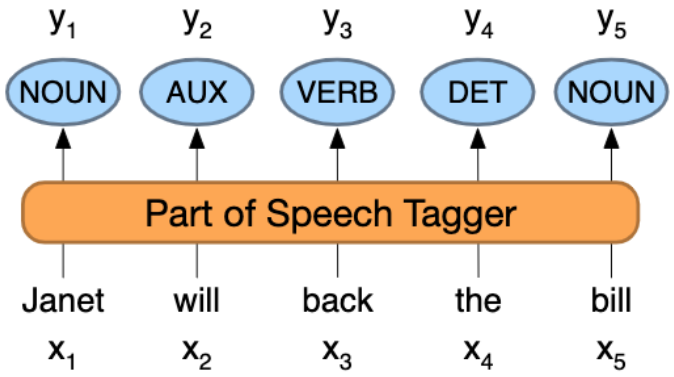


Ambiguity in POS Tagging

- “Like” can be a verb or a preposition
  - I like/VBP candy.
  - Time flies like/IN an arrow.
- “Around” can be a preposition, particle, or adverb
  - I bought it at the shop around/IN the corner.
  - I never got around/RP to getting a car.
  - A new Prius costs around/RB $25K.

Part-of-Speech Tagging

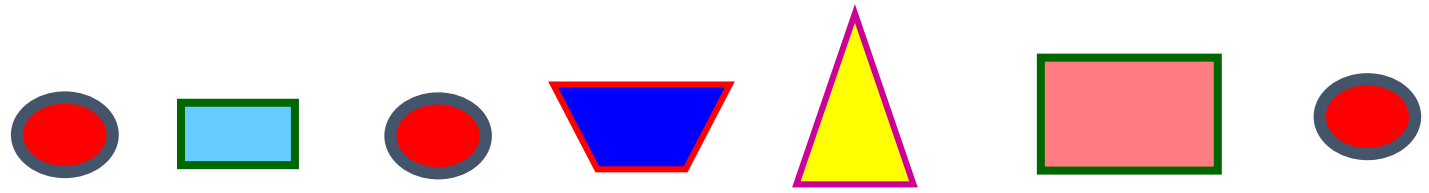
Map from sequence $x_1,...,x_n$ of words to $y_1,...,y_n$ of POS tags



Tasks in which we assign, to each word x_i in an input word sequence, a label y_i, so that the output sequence Y has the same length as the input sequence X are called sequence labeling tasks

Sequence Lebelling Tasks

- Many NLP problems can viewed as sequence labeling
- Labels of tokens are dependent on the labels of other tokens in the sequence, particularly their neighbors



Example of Sequence Labelling: Information Extraction

- Identify phrases in language that refer to specific types of entities and relations in text
- Named entity recognition is task of identifying names of people, places, organizations, etc. in text.

  people organizations places

  - Michael Dell is the CEO of Dell Computer Corporation and lives in Austin Texas
- Extract pieces of information relevant to a specific application, e.g. used car ads:

  make model year mileage price

  - For sale, 2002 Toyota Prius, 20,000 mi, $15K or best offer. Available starting July 30, 2006.

Example of Sequence Labelling: Semantic Role Labeling

- For each clause, determine the semantic role played by each noun phrase that is an argument to the verb

  agent patient source destination instrument

  - John drove Mary from Austin to Dallas in his Toyota Prius.
  - The hammer broke the window.
- Also referred to a “case role analysis,” “thematic analysis,” and “shallow semantic parsing”

Problems with Sequence Labeling as Classification

- Not easy to integrate information from the category of tokens on both sides
- Difficult to propagate uncertainty between decisions and “collectively” determine the most likely joint assignment of categories to all of the tokens in a sequence

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

Types:		WSJ		Brown	
Unambiguous	(1 tag)	44,432	**(86%)**	45,799	**(85%)**
Ambiguous	(2+ tags)	7,025	**(14%)**	8,050	**(15%)**
Tokens:					
Unambiguous	(1 tag)	577,421	**(45%)**	384,349	**(33%)**
Ambiguous	(2+ tags)	711,780	**(55%)**	786,646	**(67%)**

How difficult is POS tagging in English?

		87-tag Original Brown		45-tag Treebank Brown	
Unambiguous (1 tag)		**44,019**		**38,857**	
Ambiguous (2–7 tags)		**5,490**		**8844**	
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*)

Mrs Shaefer never got around to joining

All we goka do is go around the corner

Chateau Petrus costs around 250

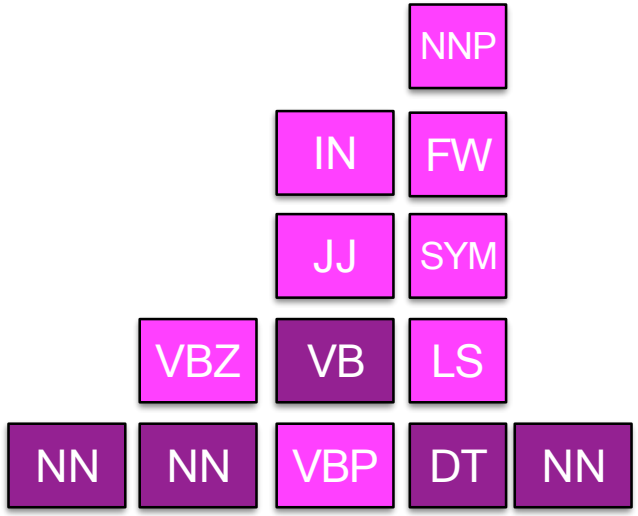
Mrs/NNP Shaefer/NNP never/RB got/VBD around/RP to/TO joining/VBG

All/DT we/PRP goka/VBN do/VB is/VBZ go/VB around/IN the/DT corner/NN

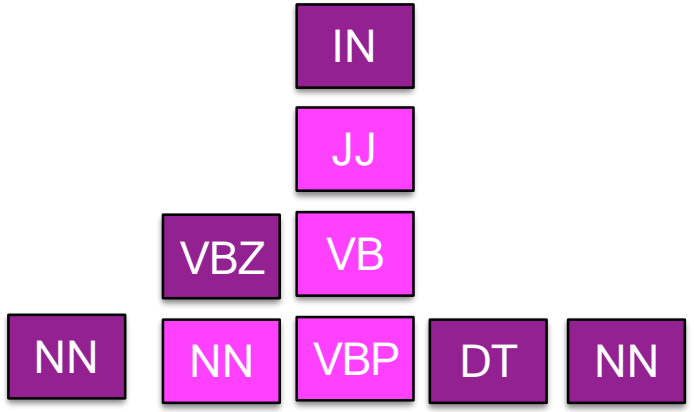
Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD

POS tagging

Labeling the tag that's correct for the context.



Fruit flies like a banana



Time flies like an arrow

(Just tags in evidence within the Penn Treebank – more are possible!)

POS tagging performance in English

- How many tags are correct? (tag accuracy)
  - About 97% (across 15 languages)
    - 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)

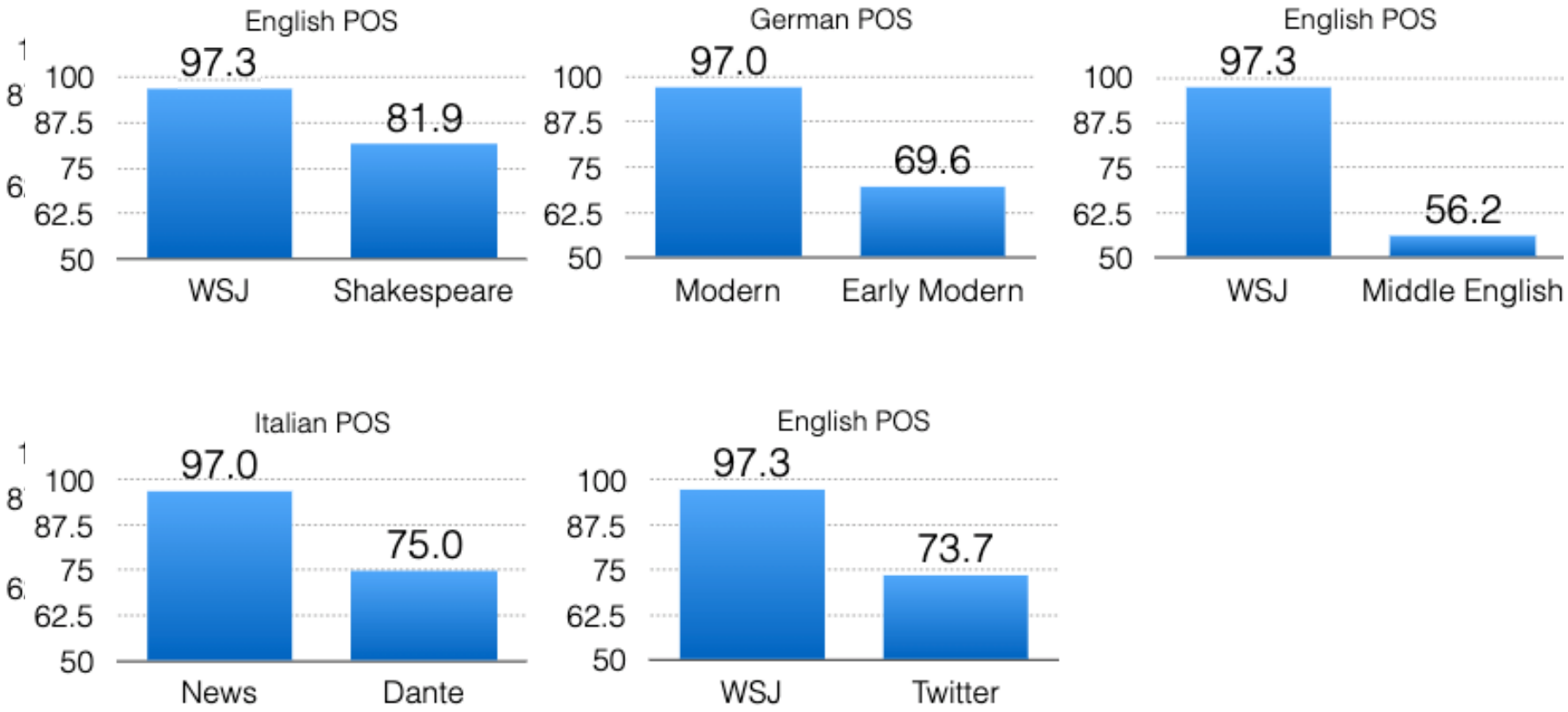
Wu,S. and Dredze, M. (2019). Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. EMNLP.

Majority class

- Pick the label each word is seen most often with in the training data

fruit	flies	like	a	banana
NN 12	VBZ 7	VB 74	FW 8	NN 3
	NNS 1	VBP 31	SYM 13	
		JJ 28	LS 2	
		IN 533	JJ 2	
			IN 1	
				DT 25820
				NNP 2

Domain Difference



Sources of information for POS tagging

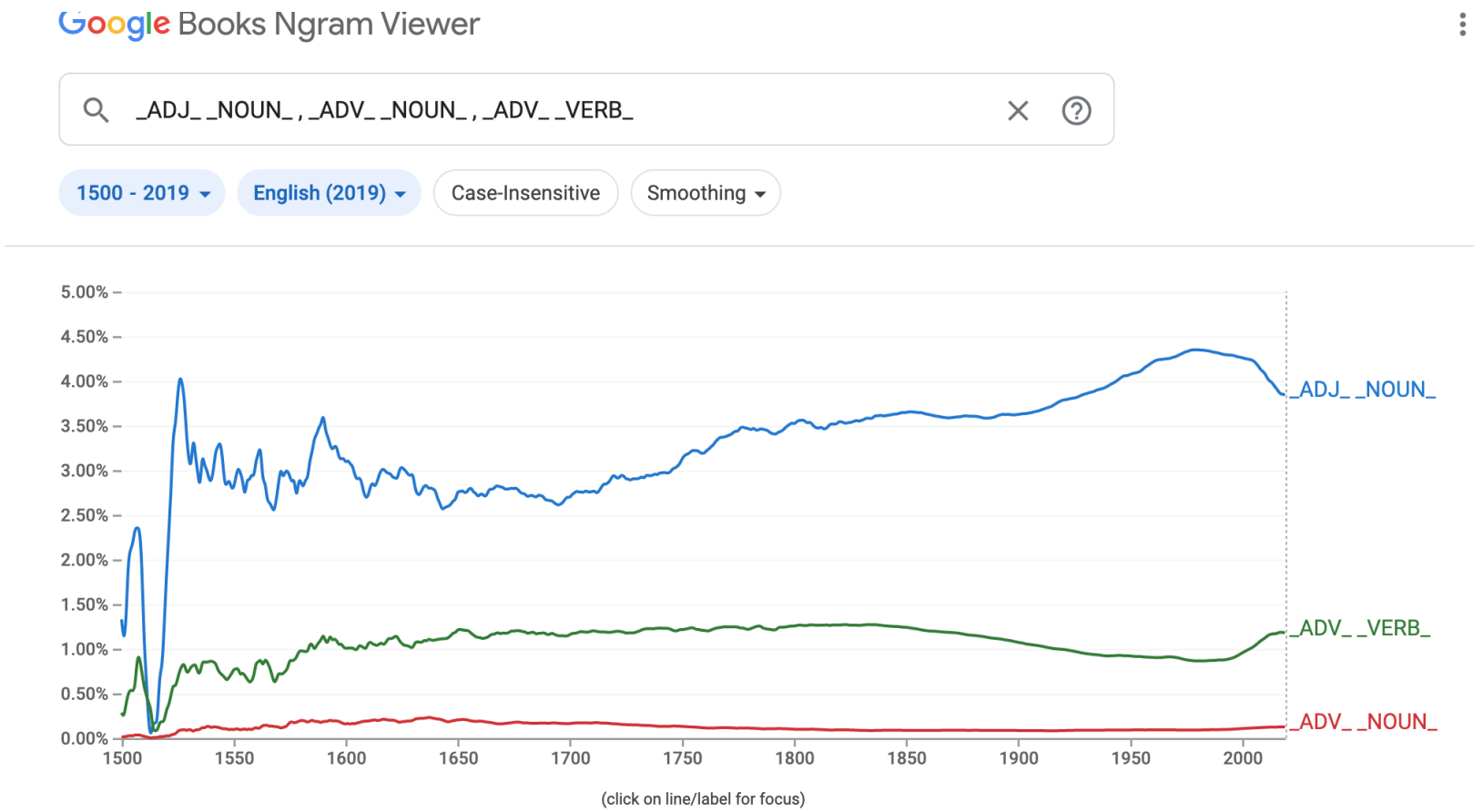
```
Janet will back the bill
 AUX/NOUN/VERB?     NOUN/VERB?
```

- Prior probabilities of word/tag
  - "will" is usually an AUX (or in other example, *man* is rarely used as a verb)
  - Some words may only be nouns, e.g. "arrow"
- Identity of neighboring words
  - "the" means the next word is probably not a verb
  - Two determiners rarely follow each other; two base form verbs rarely follow each other and determiner is almost always followed by an adjective or noun
    - Bill saw that man yesterday
    - NNP NN DT NN NN
    - VB VB(D) IN VB NN
- Morphology and wordshape:
  - Prefixes unable: un- → ADJ
  - Suffixes importantly: -ly → ADJ
  - Capitalization Janet: CAP → PROPN

FYI: Standard algorithms for POS tagging

- Rule based models:
  - human-crafted rules based on lexical and other linguistic knowledge
- 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 require a hand-labeled training set, all have about equal performance (97% on En)
- All make use of information sources we discussed
- Via human created features: HMMs and CRFs
- Via representation learning: Neural LMs

POS N-grams over time



Named Entity Recognition (NER)

Proper Nouns

- Part of speech tagging can tell us that words like Janet, University of Innsbruck, and Colorado are all proper nouns;
  - being a proper noun is a grammatical property of these words
- But viewed from a semantic perspective, these proper nouns refer to different kinds of entities:
  - Janet is a person,
  - University of Innsbruck is an organization,
  - Colorado is a location

Proper Nouns

- Proper names - another important and anciently studied linguistic category
- While parts of speech are generally assigned to individual words or morphemes, a proper name is often an entire multiword phrase, like the name “Marie Curie”, the location “New York City”, or the organization “Stanford University”
- Named entity - anything that can be referred to with a named entity proper name: a person, a location, an organization
  - although the term is commonly extended to include things that aren’t entities per se..

Named Entities

- **Named entity**, in its core usage, means anything that can be referred to with a proper name. Most common 3 or 4 tags:
  - PER (Person): "Marie Curie"
  - LOC (Location): "New York City"
  - ORG (Organization): "Stanford University"
  - MISC (Miscellaneous)

- person
- location
- organization
- time
- money
- percent
- date

7-class case

Many applications will also need to use specific entity types like proteins, genes, commercial products, or works of art

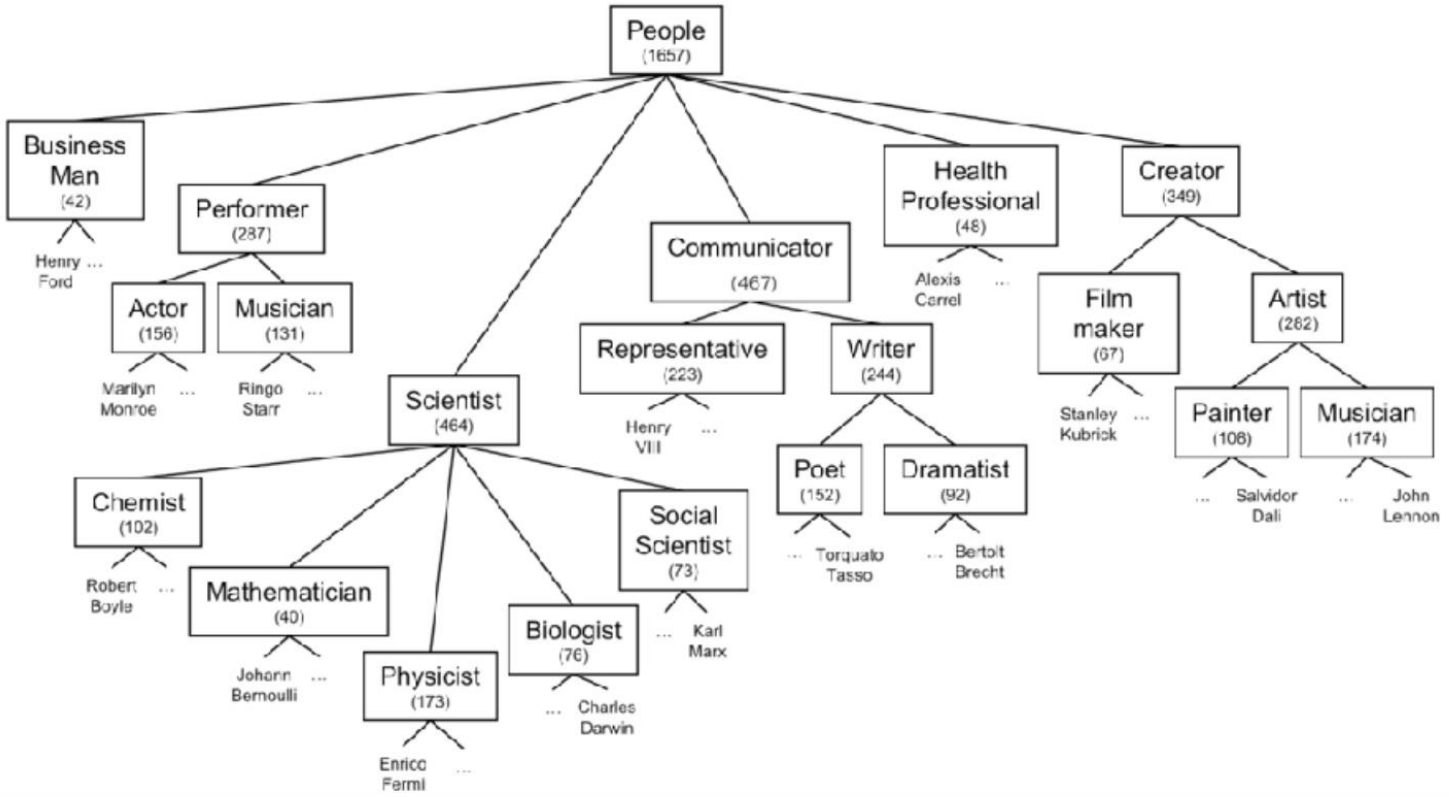
Named Entity Types

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	The **Mt. Sanitas** loop is in **Sunshine Canyon**.
Geo-Political Entity	GPE	countries, states, provinces	**Palo Alto** is raising the fees for parking.
Facility	FAC	bridges, buildings, airports	Consider the **Golden Gate Bridge**.
Vehicles	VEH	planes, trains, automobiles	It was a classic **Ford Falcon**.

These types were developed for the news domain
as part of NIST's Automatic Content Extraction (ACE) program.

Other domains (e.g. biomedical text) require different types (proteins, genes, diseases, etc.)

Fine-grained NER



Giuliano and Gliozzo (2008)

Features for NER

Lists of common names exist for many entities

- Gazetteers (place names, www.geonames.org),
- Census-derived lists of first names and surnames,
- Genes, proteins, diseases, etc.
- Company names

Such lists can be helpful, but:

… Zipf's Law: these lists are typically not exhaustive, (and the distribution of names has a long tail)

… Ambiguity: many entity names either refer to different types of entities (*Washington*: person, places named after the person), or are used to refer to different types of entity (metonymy: *Washington* as reference to the US governement)

Named Entity Recognition (NER)

- The task of **named entity recognition** (NER):

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

- Knowing if a named entity like Washington is a name of a person, a place, or a university is important to many natural language understanding tasks like:
  - question answering,
  - stance detection,
  - information extraction
  - etc.

Why NER?

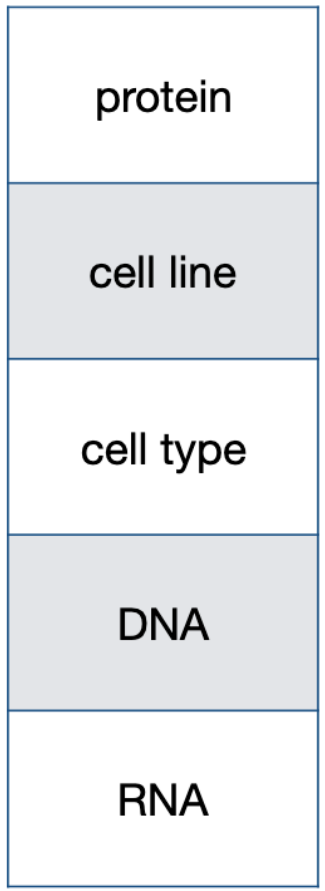
- Sentiment analysis: consumer's sentiment toward a particular company or person?
- Question Answering: answer questions about an entity?
- Information Extraction: Extracting facts about entities from text
- Linking text to information in structured knowledge sources like Wikipedia (Named Entity Linking: NEL)
- General natural language understanding tasks

NER output sample

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**].

- GENIA corpus of MEDLINE abstracts (biomedical)

We have shown that [interleukin-1]$_{\text{PROTEIN}}$ ([IL-1]$_{\text{PROTEIN}}$) and [IL-2]$_{\text{PROTEIN}}$ control [IL-2 receptor alpha (IL-2R alpha) gene]$_{\text{DNA}}$ transcription in [CD4-CD8- murine T lymphocyte precursors]$_{\text{CELL LINE}}$



http://www.aclweb.org/anthology/W04-1213

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.

- Most named entity recognition datasets have flat structure (i.e., non-hierarchical labels)

✔ [The University of California]$_{\text{ORG}}$
✖ [The University of [California]$_{\text{GPE}}$]$_{\text{ORG}}$

- Mostly fine for named entities, but more problematic for general entities:

[[John]$_{\text{PER}}$'s mother]$_{\text{PER}}$ said ...

Nested NER

named	after	the	daughter	of	a	Mattel	co-founder
						B-ORG	
					B-PER	I-PER	I-PER
		B-PER	I-PER	I-PER	I-PER	I-PER	I-PER

BIO Tagging

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

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

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

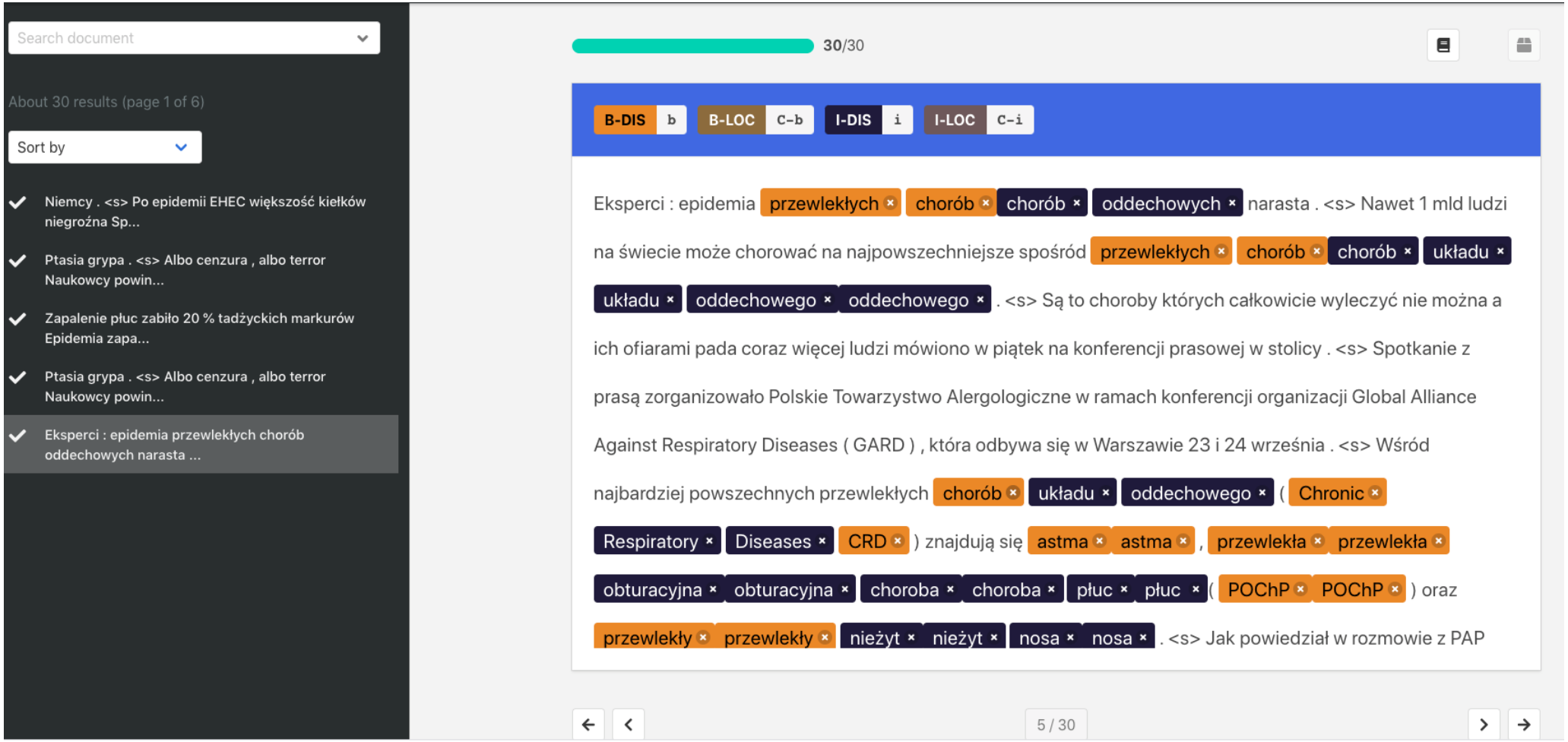
BIO tagging has the advantage that we can represent the task in the same simple sequence modeling way as part-of-speech tagging: assigning a single label y_i to each input word x_i

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

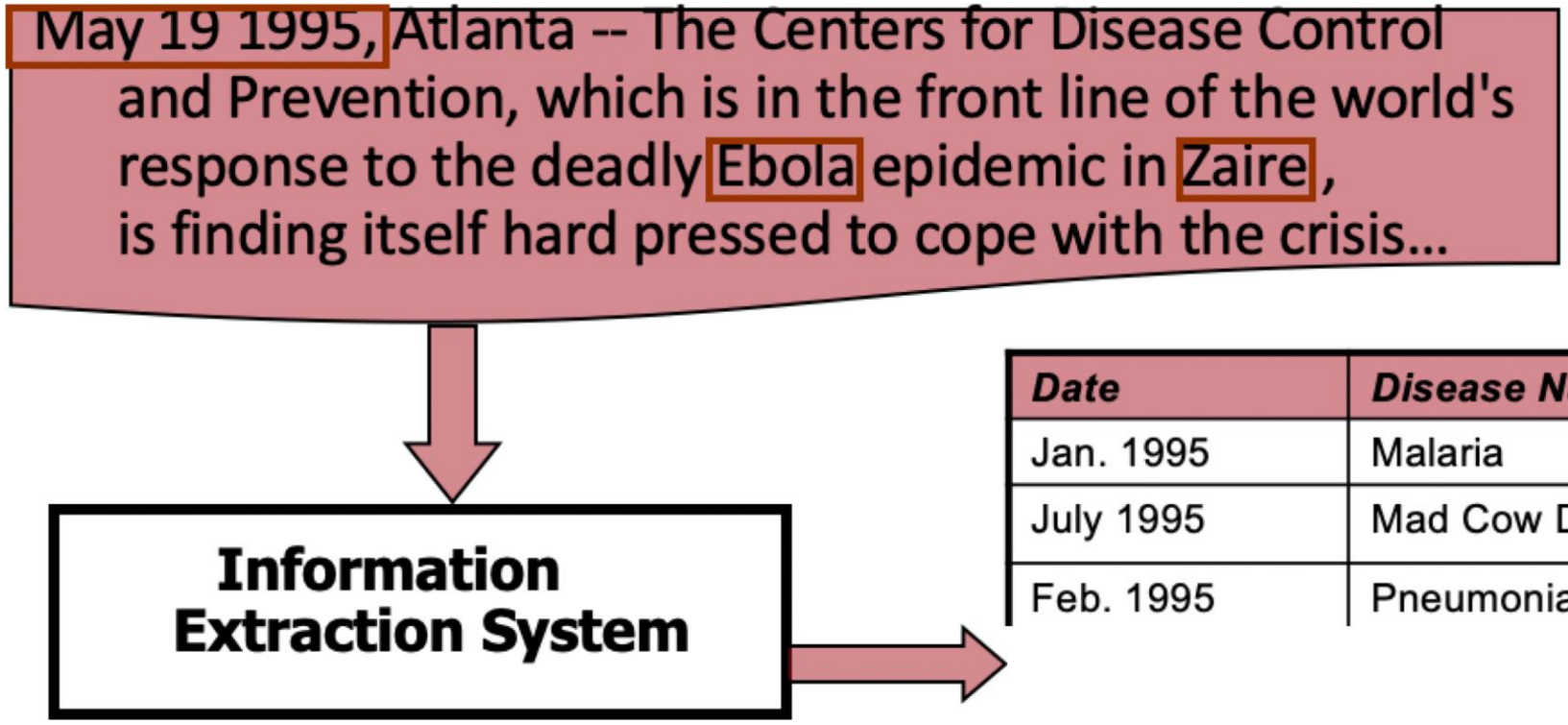
Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O

Example annotation tool: doccano – text annotation for humans



https://doccanociat.herokuapp.com/

Information Extraction: Disease Outbreaks



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

Stephen Mutuvi, Emanuela Boros, Antoine Doucet, Adam Jatowt, Gaël Lejeune, Moses Odeo: Multilingual Epidemiological Text Classification: A Comparative Study. COLING 2020: 6172-6183

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), fine-tuned

A sequence labeler (HMM, CRF, RNN, Transformer, etc.) is trained to label each token in a text with tags that indicate the presence (or absence) of particular kinds of named entities

Evaluation of the Named Entity Recognition Task

Task: Predict entities in a text

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

} Standard evaluation is per entity, *not* per token

Evaluation of NER

- Part-of-speech taggers are evaluated by the standard metric of accuracy
- Named entity recognizers are evaluated by recall, precision, and F1 measure
- Recall is the ratio of the number of correctly labeled responses to the total that should have been labeled;
- Precision is the ratio of the number of correctly labeled responses to the total labeled; and
- F-measure is the harmonic mean of the two

Evaluation of NER

- For named entities, an entity rather than a word is the unit of response. Thus, the two entities “Jane Villanueva” and “United Airlines Holding” and the non-entity “discussed” would each count as a single response
- The fact that named entity tagging has a segmentation component which is not present in tasks like text categorization or part-of-speech tagging causes some problems with evaluation
- For example, a system that labeled “Jane” but not “Jane Villanueva” as a person would cause two errors, a false positive and a false negative
- In addition, using entities as the unit of response but words as the unit of training means that there is a mismatch between the training and test conditions
  - Some other metrics (e.g., MUC scorer) give partial credit (according to complex rules)

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

Evaluation

	1	2	3	4	5	6	7
	tim	cook	is	the	CEO	of	Apple
gold	B-PER	I-PER	O	O	O	O	B-ORG
system	B-PER	O	O	O	B-PER	O	B-ORG

<start, end, type>

Precision	1/3
Recall	1/2

gold

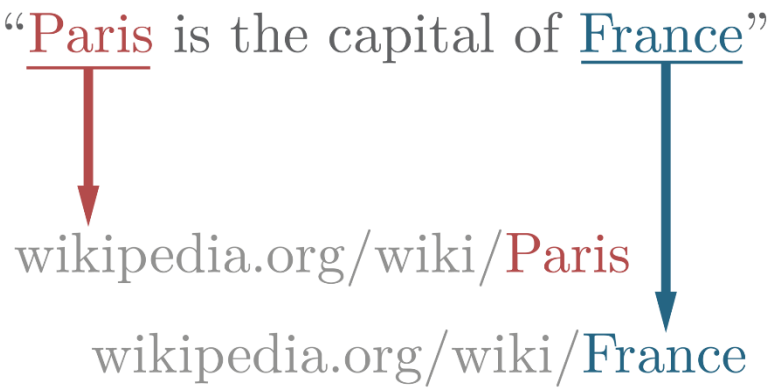
<1,2,PER>
<7,7,ORG>

system

<1,1,PER>
<5,5,PER>
<7,7,ORG>

Named entity Linking (NEL)

- Sometimes called: Entity Linking or Wikification
- The task of assigning a unique identity to entities mentioned in text
  - For example, given the sentence *"Paris is the capital of France"*, the idea is to determine that *"Paris"* refers to the city of Paris and not to Paris Hilton or any other entity that could be referred to as *"Paris"*



https://en.wikipedia.org/wiki/Entity_linking#/media/File:Entity_Linking_-_Short_Example.png

Named entity Linking (NEL)

- Task: Given a database of candidate referents, identify the correct referent for a mention in context.

Text	True wikipedia page
Hornets owner **Michael Jordan** thinks having one or two "superteams" is a detriment to the NBA because the other 28 teams "are going to be garbage."	`wiki/Michael_Jordan`
In 2001, **Michael Jordan** and others resigned from the Editorial Board of *Machine Learning*.	`wiki/Michael_I._Jordan`
The stars are aligning for leading man **Michael Jordan**, who just signed on for a new film, according to Variety.	`wiki/Michael_B._Jordan`
Michael Jordan played in 1,072 regular-season games in his 15-season career	`wiki/Michael_Jordan`

Michael Jordan (disambiguation)

From Wikipedia, the free encyclopedia

Michael Jordan (born 1963) is an American basketball player.

Michael or **Mike Jordan** may also refer to:

People [edit]

Sports [edit]

- Michael Jordan (footballer) (born 1986), English goalkeeper
- Mike Jordan (racing driver) (born 1958), English racing driver
- Mike Jordan (baseball, born 1863) (1863–1940), baseball player
- Mike Jordan (cornerback) (born 1992), American football cornerback
- Michael-Hakim Jordan (born 1977), American professional basketball player
- Michal Jordán (born 1990), Czech ice hockey player

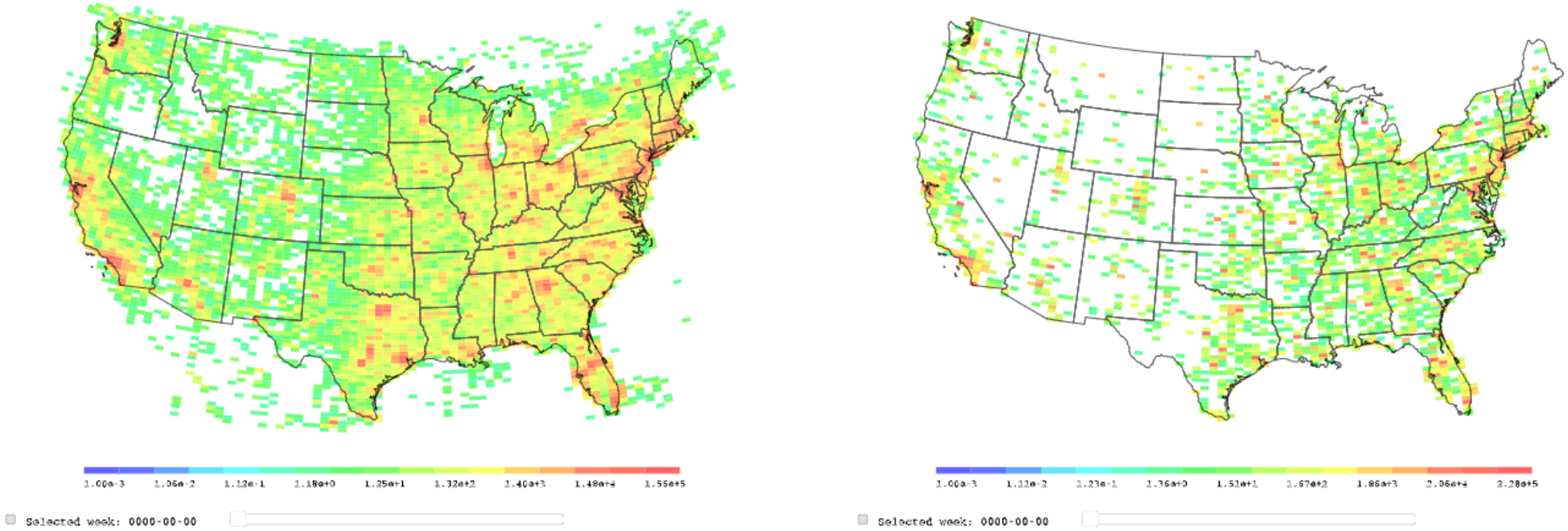
Other people [edit]

- Michael B. Jordan (born 1987), American actor
- Michael Jordan (insolvency baron) (born 1931), English businessman
- Michael Jordan (Irish politician), Irish Farmers' Party TD from Wexford, 1927–1932
- Michael I. Jordan (born 1956), American researcher in machine learning and artificial intelligence
- Michael H. Jordan (1936–2010), American executive for CBS, PepsiCo, Westinghouse
- Michael Jordan (mycologist), English mycologist

Tagme

https://sobigdata.d4science.org/web/tagme/demo

Research Example of Analysis focused on detected locations



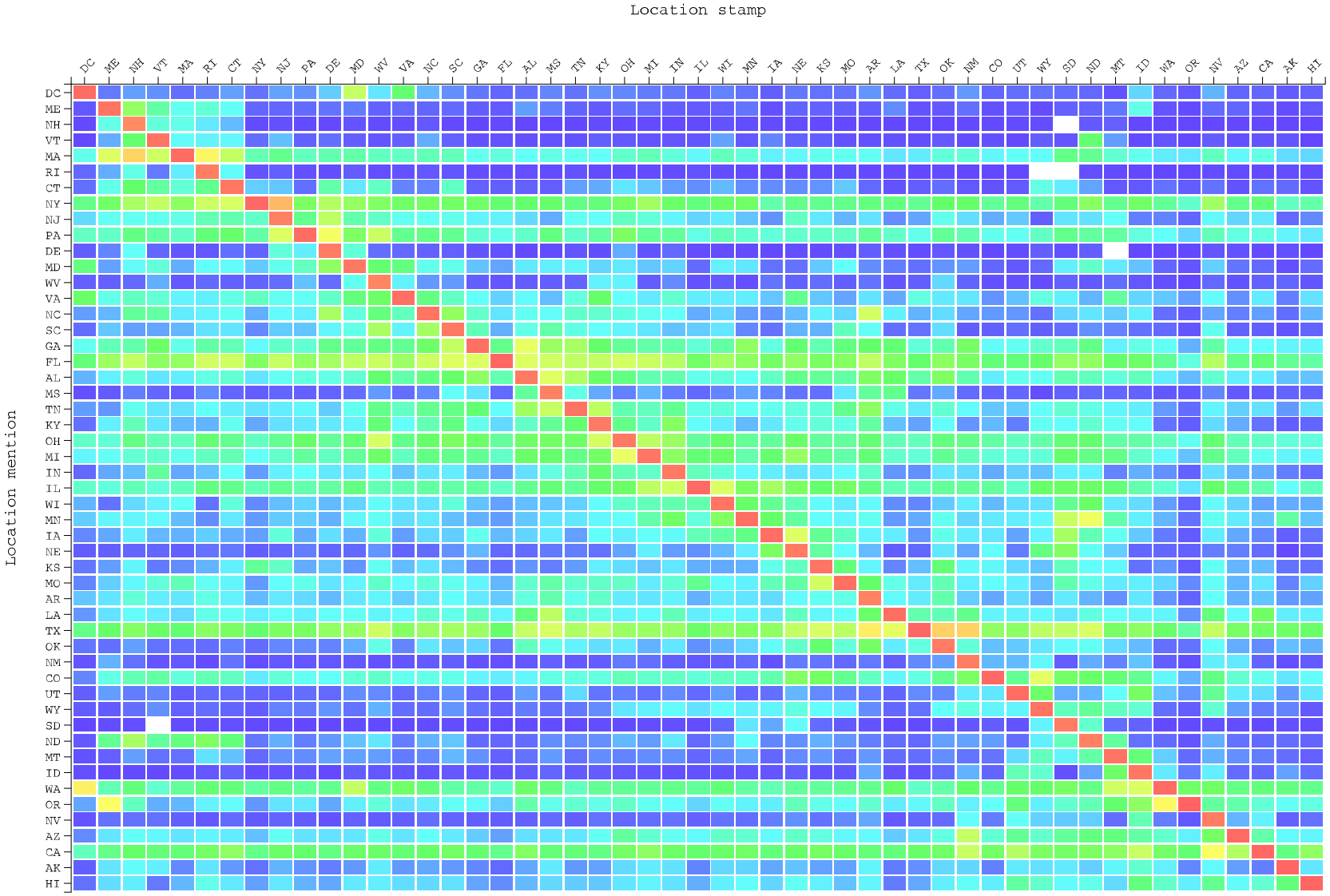
(a) Tweets mapped to their location stamp [45]

(b) Tweets mapped to their location mention [46]

Figure 6: Geographic distribution of tweets according to their location stamp and mention

Study of location mentions in USA in tweets within USA in 2013

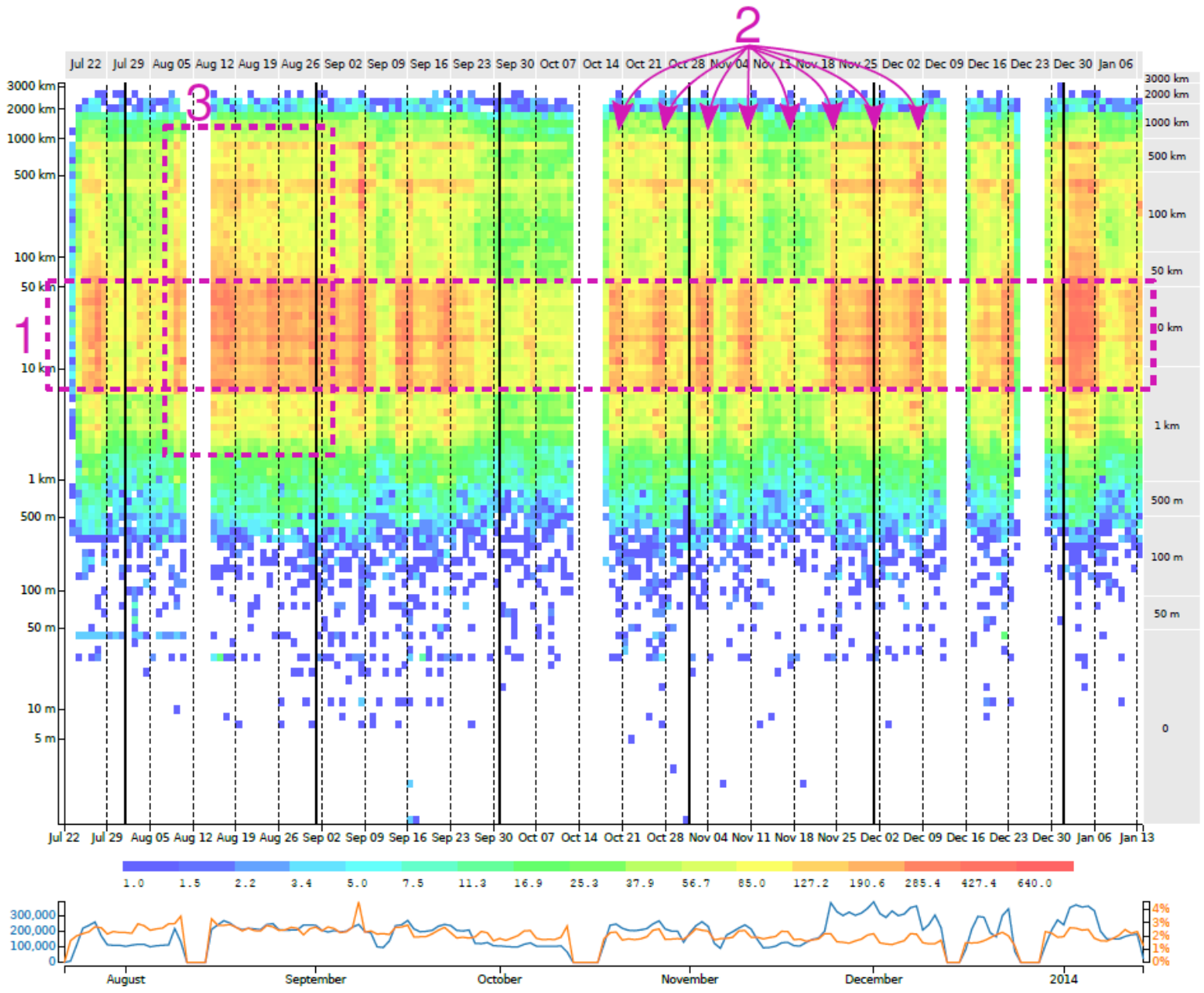
Émilien Antoine, Adam Jatowt, Shoko Wakamiya, Yukiko Kawai, Toyokazu Akiyama: Portraying Collective Spatial Attention in Twitter. KDD 2015: 39-48



Popularity of a US state listed in a row among states from where tweets originate

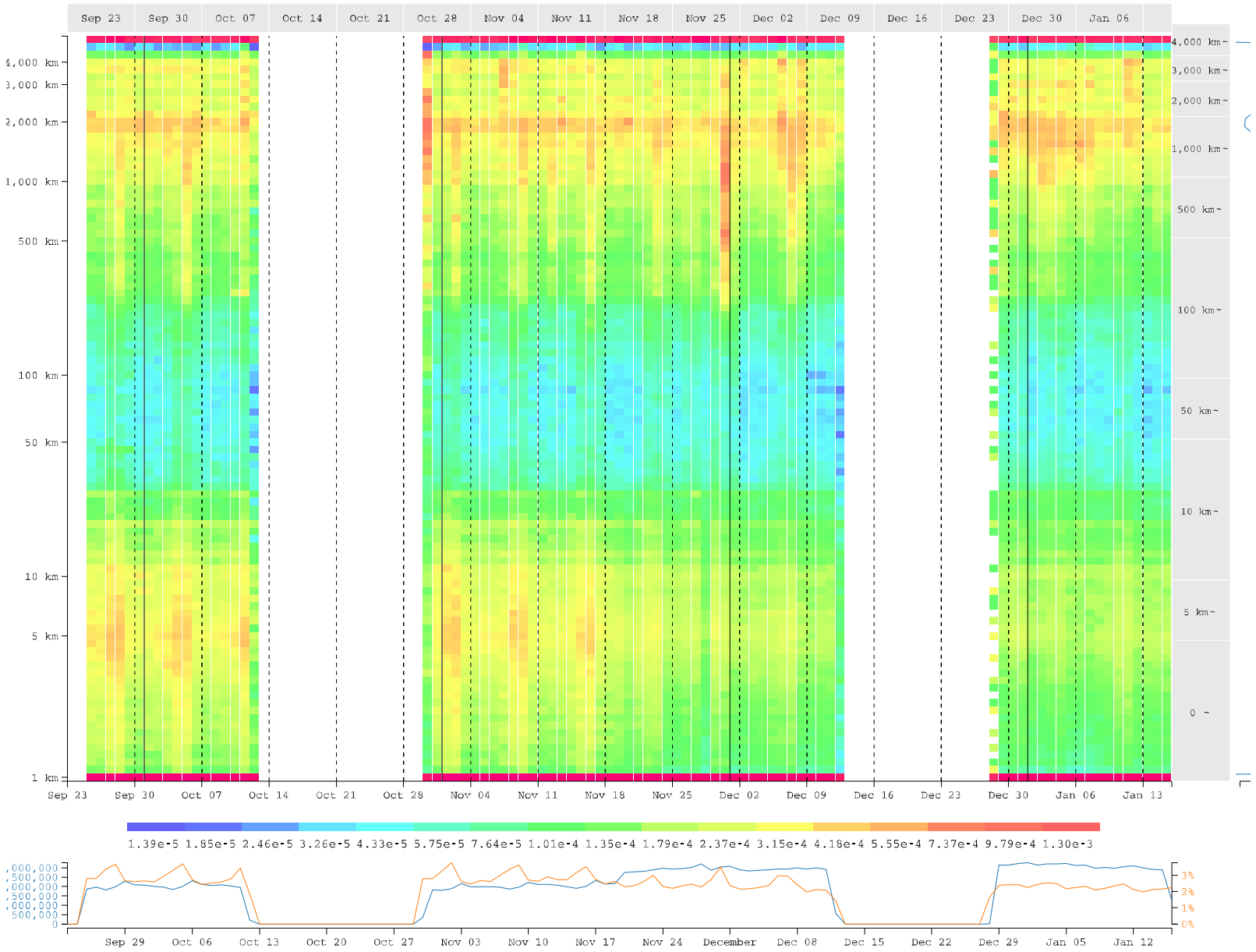
Émilien Antoine, Adam Jatowt, Shoko Wakamiya, Yukiko Kawai, Toyokazu Akiyama: Portraying Collective Spatial Attention in Twitter. KDD 2015: 39-48

Japan



Émilien Antoine, Adam Jatowt, Shoko Wakamiya, Yukiko Kawai, Toyokazu Akiyama: Portraying Collective Spatial Attention in Twitter. KDD 2015: 39-48

USA

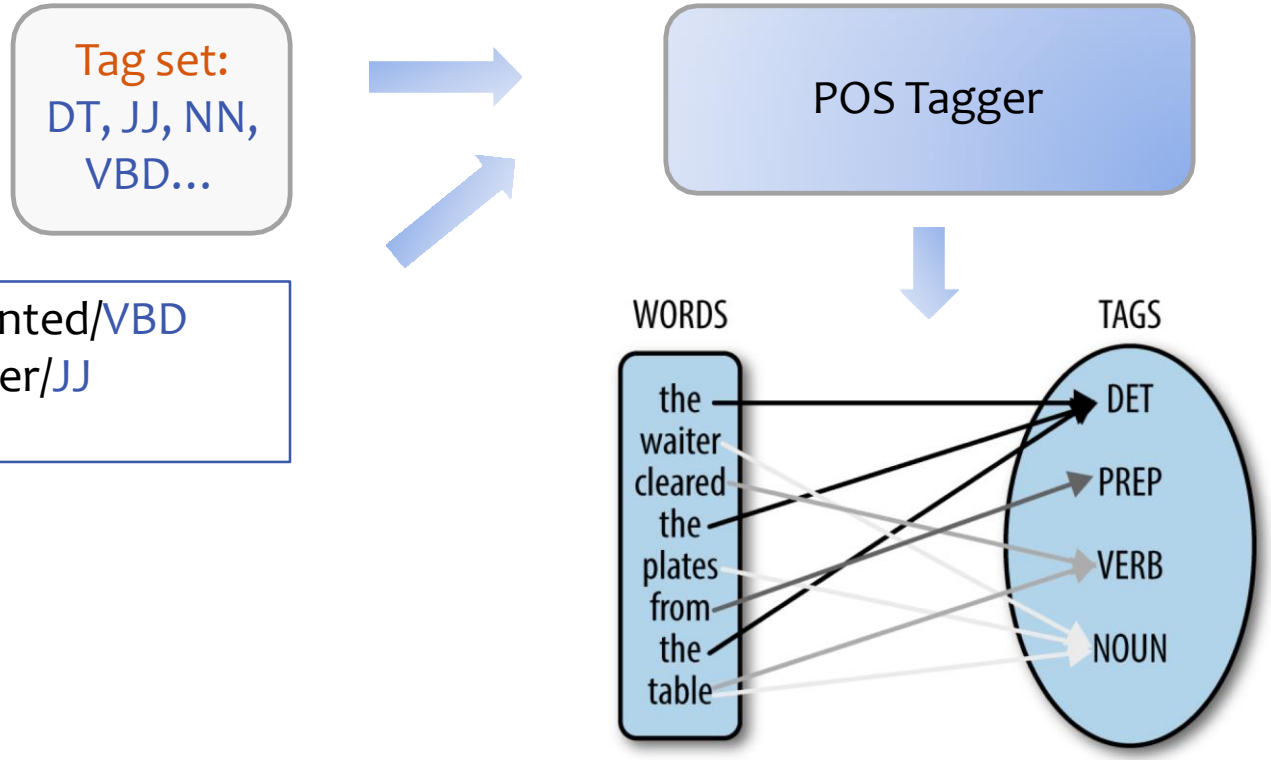


Émilien Antoine, Adam Jatowt, Shoko Wakamiya, Yukiko Kawai, Toyokazu Akiyama: Portraying Collective Spatial Attention in Twitter. KDD 2015: 39-48

Hidden Markov Models (POS tagging example)

Building a POS Tagger

- Supervised Learning
  - Assume linguists have annotated several examples



Hidden Markov Model

- HMM: a sequence model
  - Probabilistic generative model for sequences
  - Assumes an underlying set of ***hidden*** (unobserved, latent) states in which the model can be (e.g. parts of speech)
  - Assumes probabilistic transitions between states over time (e.g. transition from POS to another POS as sequence is generated)
  - Assumes a ***probabilistic*** generation of tokens from states (e.g. words generated for each POS)
  - A Hidden Markov Model is an extension of a Markov chain in which the input symbols are not the same as the states
    - For usual Markov chains, the output symbols are the same as the states
    - This means we don't know which state we are in (states are not observable, only output symbols)

- What is the most likely sequence of tags for the given sequence of words **w**?

How Can We Predict the POS Tags?

Two types of information are useful

- Relations between words and tags
- Relations between tags and tags
  - DT NN, DT JJ NN …

Statistical POS Tagging

- What is the most likely **sequence of tags** for the given **sequence of words w**?

$$
\begin{aligned}
\underset{\mathbf{t}}{\operatorname{argmax}} P(\mathbf{t}|\mathbf{w}) &= \underset{\mathbf{t}}{\operatorname{argmax}} \frac{P(\mathbf{t},\mathbf{w})}{P(\mathbf{w})} \\
&= \underset{\mathbf{t}}{\operatorname{argmax}} P(\mathbf{t},\mathbf{w}) \\
&= \underset{\mathbf{t}}{\operatorname{argmax}} P(\mathbf{t})P(\mathbf{w}|\mathbf{t})
\end{aligned}
$$

Statistical POS Tagging

- What is the most likely sequence of tags for the given sequence of words **w**?

$$P(\text{DT JJ NN} \mid \text{a smart dog}) = P(\text{DD JJ NN a smart dog}) / P(\text{a smart dog})$$
$$\propto P(\text{DD JJ NN a smart dog})$$
$$= P(\text{DD JJ NN})\, P(\text{a smart dog} \mid \text{DD JJ NN})$$

Transition Probability: First-order Markov Model

- Joint probability $P(t, \mathrm{w}) = P(t)P(\mathrm{w}|t)$

$$P(\boldsymbol{t}) = P(t_1, t_2, \ldots t_n)$$
$$= P(t_1)P(t_2 \mid t_1)P(t_3 \mid t_2, t_1) \ldots P(t_n \mid t_1 \ldots t_{n-1})$$
$$\sim \mathrm{P}(\mathrm{t}_1)\mathrm{P}(\mathrm{t}_2 \mid t_1)P(t_3 \mid t_2) \ldots P(t_n \mid t_{n-1})$$
$$= \Pi_{i=1}^{n} P(t_i \mid t_{i-1})$$

Markov Assumption

- Bigram model over POS tags!
  (similarly, we can define an n-gram model over POS tags, usually called high-order HMM)

Sequences

Most common tag bigrams in Penn Treebank training

DT	NN	41909
NNP	NNP	37696
NN	IN	35458
IN	DT	35006
JJ	NN	29699
DT	JJ	19166
NN	NN	17484
NN	,	16352
IN	NNP	15940
NN	.	15548
JJ	NNS	15297
NNS	IN	15146
TO	VB	13797
NNP	,	13683
IN	NN	11565

Emission Probability

- Joint probability $P(t, \mathrm{w}) = P(t)P(\mathrm{w}|t)$
- Assume words depend only on their POS-tags

$$P(\boldsymbol{w}|\boldsymbol{t}) \sim P(w_1 \mid t_1)P(w_2 \mid t_2) \ldots P(w_n \mid t_n)$$
$$= \Pi_{i=1}^{n} P(w_i|t_i)$$

Independence Assumption

i.e., P(a smart dog | DD JJ NN)

= P(a | DD) P(smart | JJ) P(dog | NN)

Putting them Together

- Joint probability $P(t, \mathrm{w}) = P(t)P(\mathrm{w}|t)$

$$P(\boldsymbol{t}, \boldsymbol{w})$$
$$= \mathrm{P}(\mathrm{t}_1)\mathrm{P}(\mathrm{t}_2 \mid t_1)P(t_3 \mid t_2) \ldots P(t_n \mid t_{n-1})P(w_1 \mid t_1)P(w_2 \mid t_2) \ldots P(w_n \mid t_n)$$
$$= \Pi_{i=1}^{n} P(w_i|t_i)P(t_i \mid t_{i-1})$$

e.g., P(a smart dog , DD JJ NN)

= P(DD | start) P(JJ | DD) P(NN | JJ)

P(a | DD) P(smart | JJ) P(dog | NN)

Putting them Together

Two independence assumptions:

- Approximate p(**t**) by a bi(or N)-gram model
  - Markov assumption: $P(q_i|q_1 \dots q_{i-1}) = P(q_i|q_{i-1})$
- Assume each word depends only on its POS tag (q denotes state/tag here and o is observation/word)
  - Output independence: $P(o_i|q_1 \dots q_i, o_1 \dots o_i \dots o_T) = P(o_i|q_i)$

Hidden Markov Models (Formal)

- States T = $t_1, t_2 \ldots t_N$
- Observations O= $o_1, o_2 \ldots o_M$
  - Each observation is a symbol from a vocabulary V = $\{v_1, v_2, \ldots v_V\}$
- Transition probabilities
  - Transition probability matrix A = $\{a_{ij}\}$
- Observation likelihoods
  - Output probability matrix B = $\{b_i(o_t)\}$
- Initial probability vector π

$$\pi_i = P(t_1 = i) \;\; 1 \leq i \leq N$$

The Components of an HMM Tagger

- An HMM has two components: A and B Probabilities
- **A**:
  - Tag transition probabilities $p(t_i|t_{i-1})$
  - The prob. of a tag occurring given the previous tag
  - Compute the MLE by counting: $p(t_i|t_{i-1}) = \frac{C(t_{i-1},t_i)}{C(t_{i-1})}$
  - Example:
    - Modal verb **will** and a verb in the base form, a VB, like **race** (WSJ corpus)

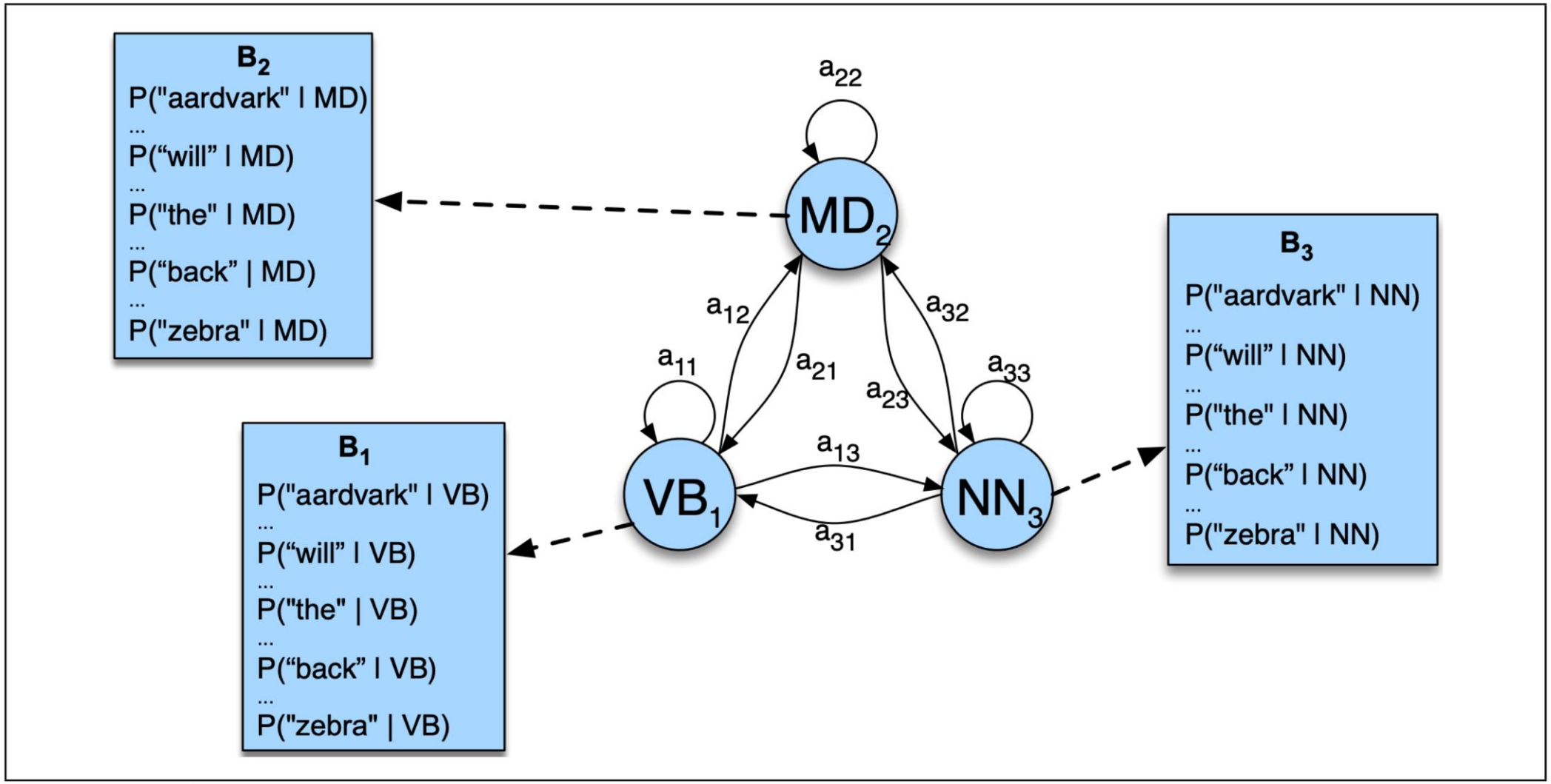
$$P(VB|MD) = \frac{C(MD,VB)}{C(MD)} = \frac{10471}{13124} = .80$$

The Components of an HMM Tagger

- An HMM has two components: A and B Probabilities
- **B**:
  - Emission probabilities $p(w_i|t_i)$
  - Given a tag (e.g., MD), that it will be associated with a given word (e.g., **will**)
  - Compute the MLE by counting: $p(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$
  - Example:
    - Modal verb MD and the word **will** (WSJ corpus)

$$P(will|MD) = \frac{C(MD, will)}{C(MD)} = \frac{4046}{13124} = .31$$

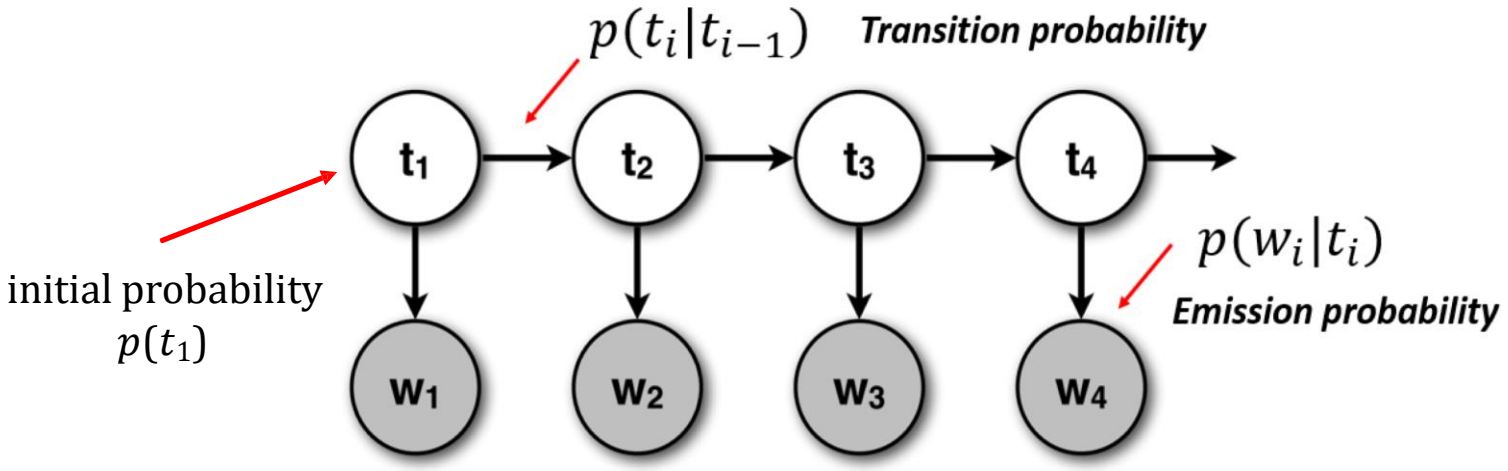
The Components of an HMM Tagger



An illustration of the two parts of an HMM representation: the A transition probabilities used to compute the prior probability, and the B observation likelihoods that are associated with each state, one likelihood for each possible observation word

HMM Tagging as Decoding

What is the most likely sequence of tags for the given sequence of words w?



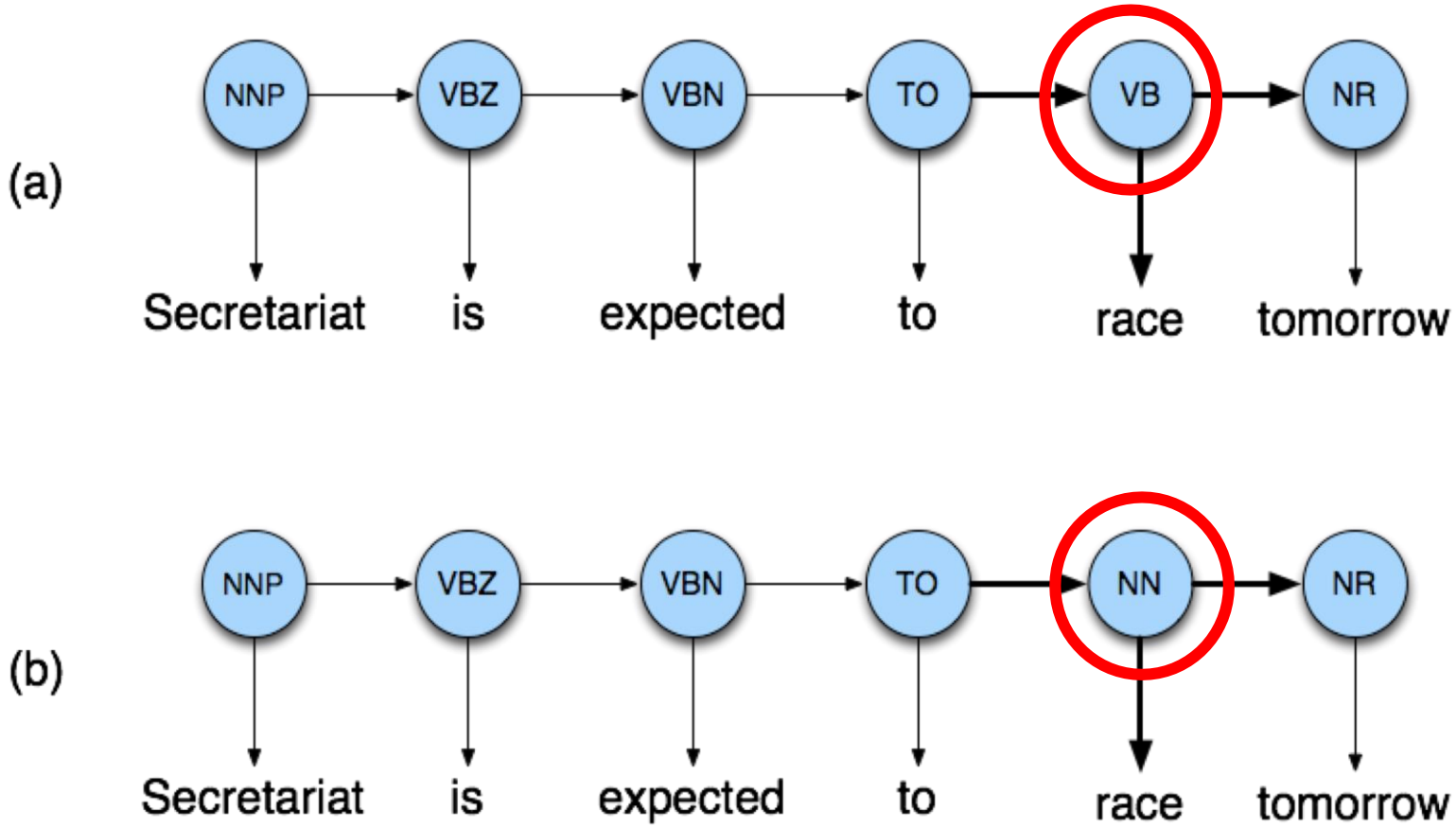
$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n) \approx \underset{t_1 \ldots t_n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i|t_i)}^{\text{emission}} \overbrace{P(t_i|t_{i-1})}^{\text{transition}}$$

Most probable tag sequence for bigram tagger

Example: The word “Race”

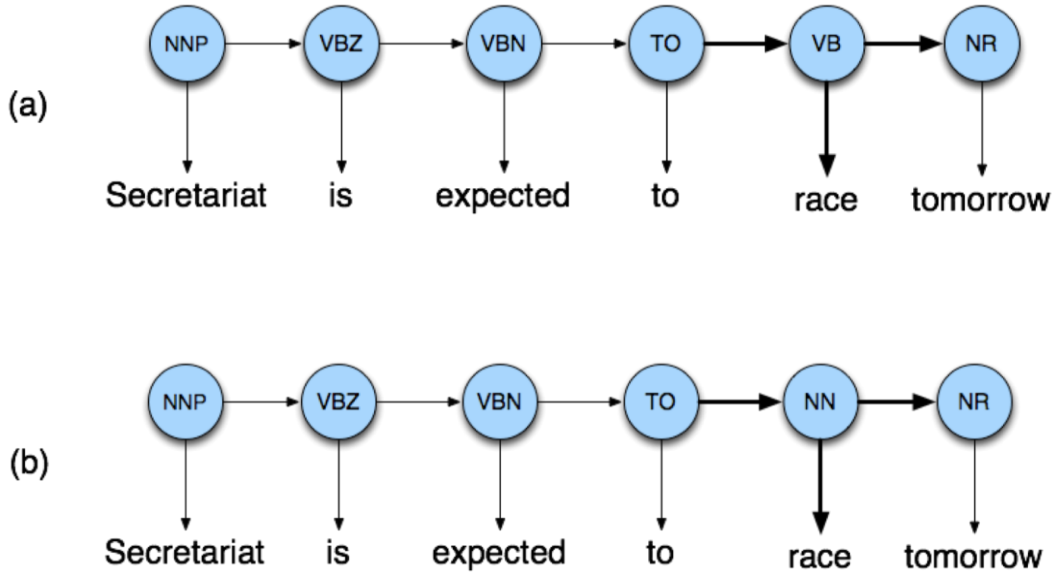
- Secretariat/NNP is/VBZ expected/VBN to/TO **race**/VB tomorrow/NR
- People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT **race**/NN for/IN outer/JJ space/NN
- How do we pick the right tag?

Disambiguating “Race” verb

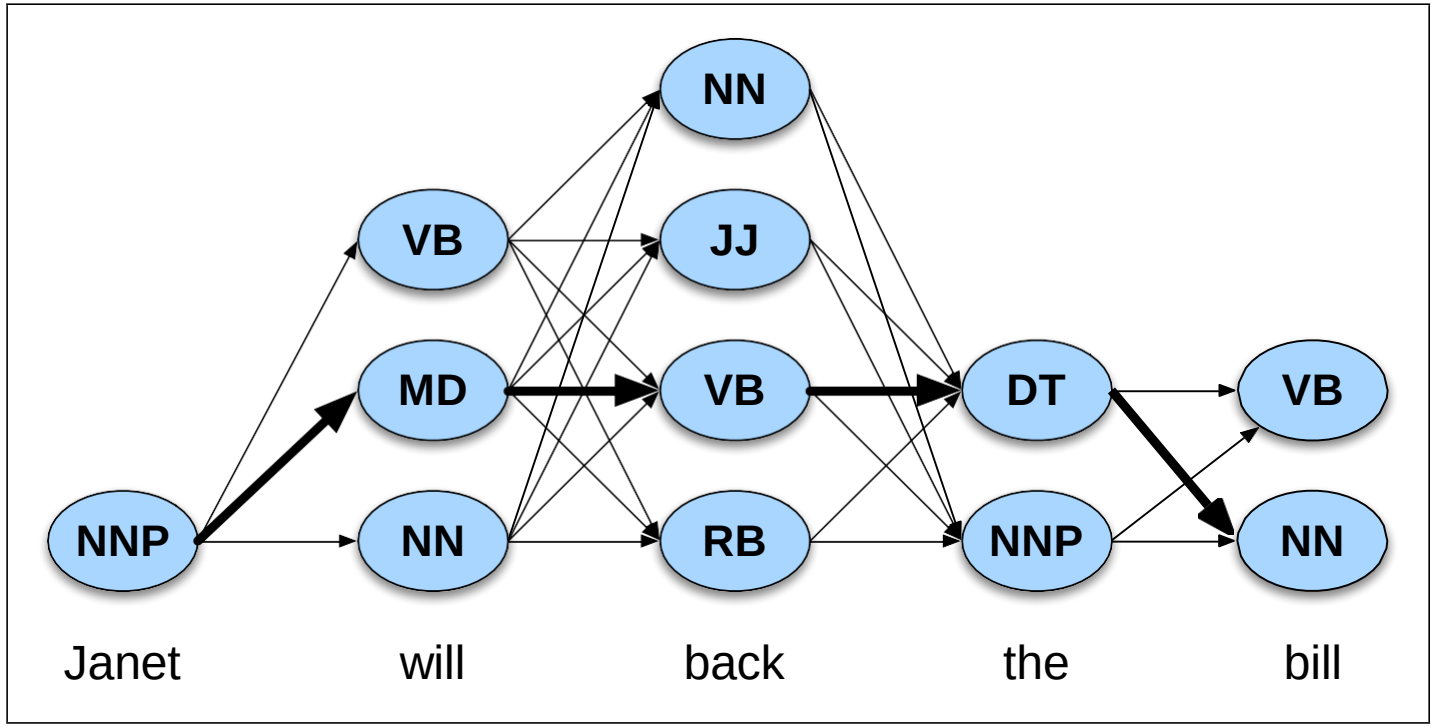


Disambiguating “Race”

- P(NN|TO) = .00047
- P(VB|TO) = .83
- P(race|NN) = .00057
- P(race|VB) = .00012
- P(NR|VB) = .0027
- P(NR|NN) = .0012
- P(VB|TO)P(NR|VB)P(race|VB) = .00000027
- P(NN|TO)P(NR|NN)P(race|NN)=.00000000032
- So we (correctly) choose the verb



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



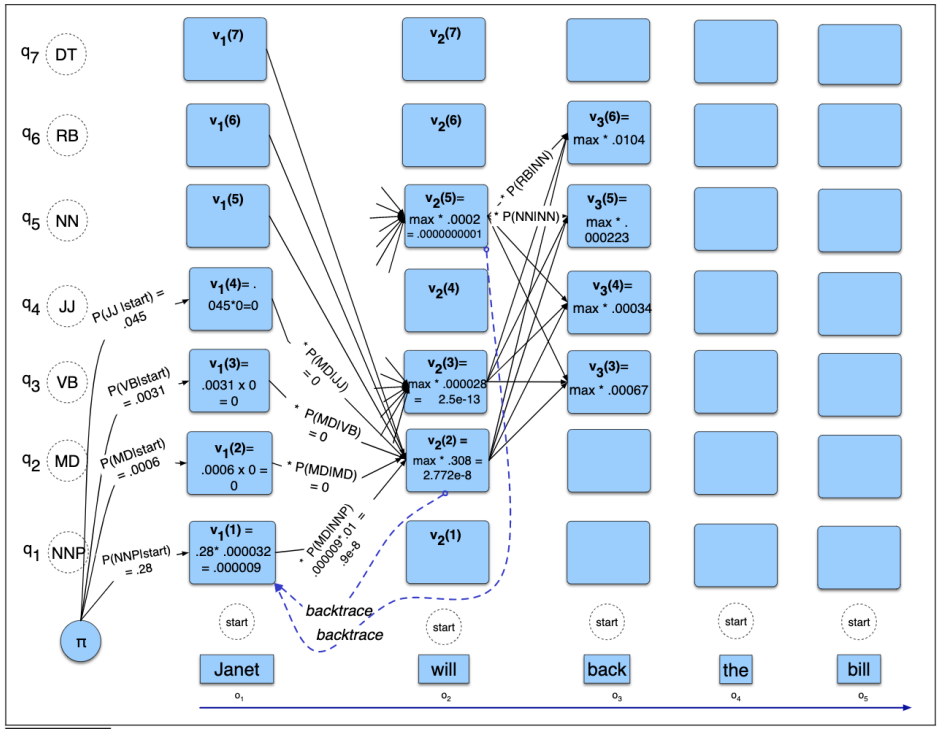
A schematic of the tagging task for the sample sentence, showing the ambiguities for each word and the correct tag sequence as the highlighted path through the hidden states.

Viterbi Algorithm

- We could just enumerate all paths given the input and use the model to assign probabilities to each
- Not very good idea..
- Luckily dynamic programming helps us here
- Viterbi algorithm resembles the dynamic programming for minimum edit distance that we studied before

Viterbi Alg. Summary

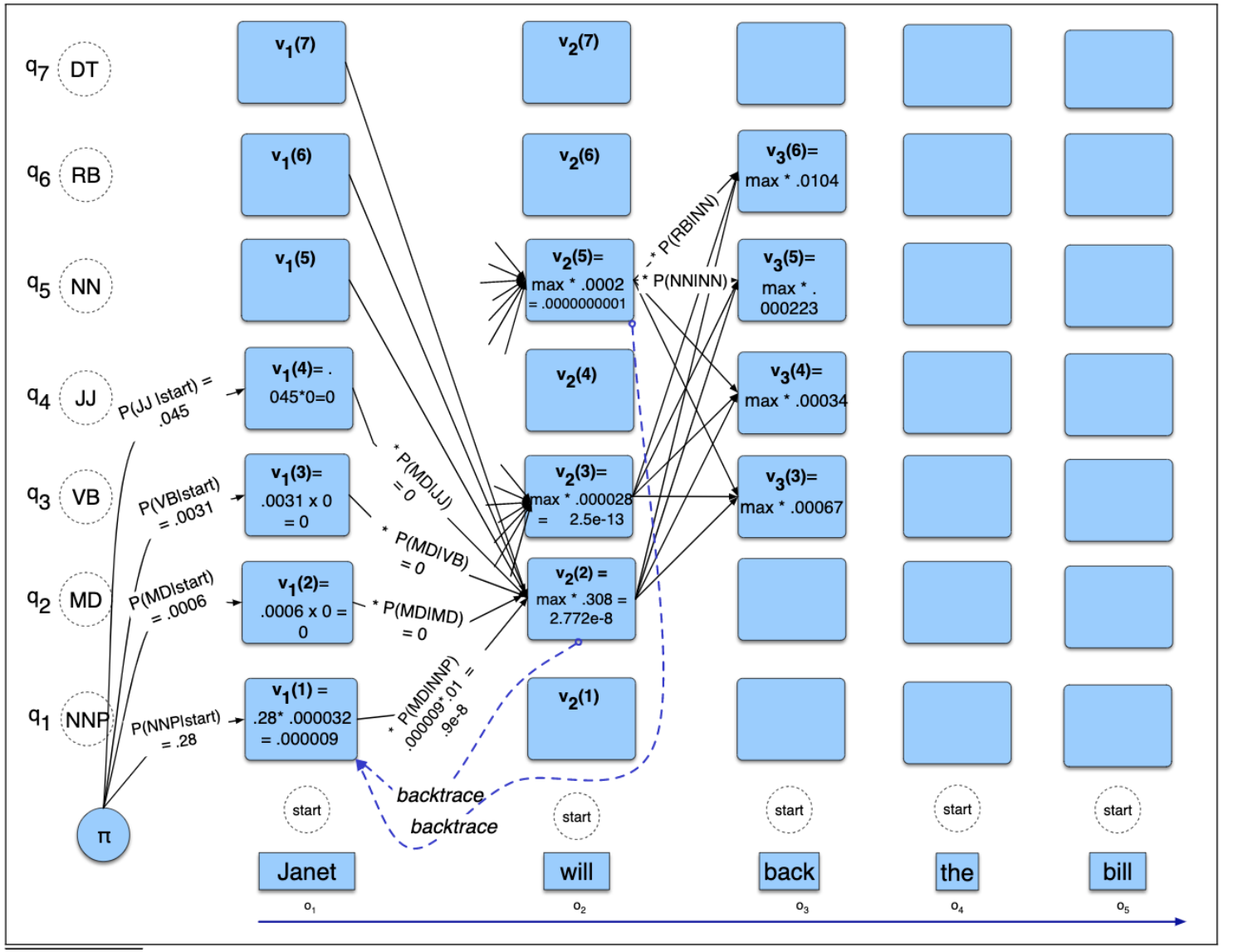
- Create an array
  - With columns corresponding to inputs
  - Rows corresponding to possible states
- Sweep through the array in one pass filling the columns left to right using transition probabilities and observations probabilities
- Dynamic programming key is that we need only store the MAX probability path to each cell (not all paths)



Viterbi probability

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$$

$v_{t-1}(i)$	the **previous Viterbi path probability** from the previous time step
a_{ij}	the **transition probability** from previous state q_i to current state q_j
$b_j(o_t)$	the **state observation likelihood** of the observation symbol o_t given the current state j



The first few entries in the individual state columns for the Viterbi algorithm (data in next slide). Each cell keeps the probability of the best path so far and a pointer to the previous cell along that path.

Columns 1 and 2 are only filled out; to avoid clutter most cells with value 0 are left empty. After the cells are filled in, backtracing from the end state, we should be able to reconstruct the correct state sequence NNP MD VB DT NN

	NNP	MD	VB	JJ	NN	RB	DT
<*s*>	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

The A transition probabilities $P(t_i|t_{i-1})$ computed from the WSJ corpus with-out smoothing. Rows are labeled with the conditioning event; thus P(VB|MD) is 0.7968

	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0	0
NN	0	0.000200	0.000223	0	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

Observation likelihoods B computed from the WSJ corpus without smoothing, simplified slightly.

Smoothing

- One solution: add a little probability mass to every element.

maximum likelihood estimate

$$P(x_i \mid y) = \frac{n_{i,y}}{n_y}$$

$n_{i,y}$ = count of word i in class y
n_y = number of words in y
V = size of vocabulary

smoothed estimates

$$P(x_i \mid y) = \frac{n_{i,y} + a}{n_y + Va}$$

same α for all x_i

$$P(x_i \mid y) = \frac{n_{i,y} + a_i}{n_y + \sum_{j=1}^{V} a_j}$$

possibly different α for each x_i

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

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

Viterbi algorithm for finding the optimal sequence of tags. Given an observation sequence and an HMM λ= (A,B), the algorithm returns the state path through the HMM that assigns maximum likelihood to the observation sequence