Faculty for Informatics

Technical University of Munich

Natural Language Processing IN2361

PD Dr. Georg Groh

Social Computing Research Group

Chapter 8
Part-of-Speech Tagging

- content is based on [1]
- certain elements (e.g. equations or tables) were taken over or taken over in a modified form from [1]
- citations of [1] or from [1] are omitted for legibility
- errors are fully in the responsibility of Georg Groh
- BIG thanks to Dan and James for a great book!

Part-of-Speech-Tags

- Parts-of-Speech (POS, word classes, syntactic categories)

- Examples: noun, pronoun, verb, adjective, ….

- important for
  - language models (“nouns are preceded by determines or adjectives”),
  - information extraction tasks such as Named Entity Recognition and Classification,
  - stemming,
  - auto-summarization,
  - pronunciation (e.g. CONtent vs conTENT)
  - etc.

Part-of-Speech-Tags

- POS: based on not primarily semantic categories (adjective ←→ property of smth) but rather
  - syntactic categories / functions (e.g. distributional properties (which other words usually in neighborhood)) and
  - morphological categories/functions (e.g. to carry similar suffixes)

- closed class (function words (e.g. *of, it*); fixed members (e.g. prepositions)) vs.
  open class (nouns, verbs, adjectives, adverbs; e.g. new nouns are continually created)

Part-of-Speech-Tags

- Nouns:
  - occur with determiners (*a goat, its bandwidth*)
  - can take possessives (*husband's house*)
  - may occur in plural (*goats, hounds*)
  - Proper Nouns: specific entities, no *the* (*Regina, IBM, Colorado*) (usually capitalized)
  - Common Nouns:
    - Count Nouns: *one goat, two goats*
    - Mass Nouns: *snow, salt, communism*

- Verbs:
  - ←→ actions, processes, smth. dynamic,…
  - may be inflected: *eat, eats, eating, eaten*

- Adjectives
  - ←→ properties, qualities,…
  - *beautiful, tall, small*

Part-of-Speech-Tags

- Adverbs:
  - modify something: ***Unfortunately****, John walked home* ***extremely slowly yesterday***
  - directional adverbs / locative adverbs: *home, here, downhill*
  - degree adverbs: *extremely, very, somewhat*
  - manner adverbs: *slowly, slinkily, delicately*
  - temporal adverbs: *yesterday, Monday*

- Prepositions:
  - occur before noun phrases: ***by*** *the house,* ***on*** *time,* ***with*** *gusto,* ***at*** *the gate*
  - indicate spatial, or temporal, or other relations

- Particle:
  - occur with verbs: *hand the paper* ***over****, throw the ball* ***at***
  - together with verb: phrasal verb (with non-compositional meaning): *turn down == reject, rule out == eliminate, go on == continue*

Part-of-Speech-Tags

- Determiners:
  - especially articles: definite: *the*; indefinite: *a, an*
  - also*: this, that, …*

- Conjunctions:
  - join phrases, sentences, clauses
  - Coordinating conjunctions*: and, or*
  - Subordinating conjunctions (Complementizers)*: I thought **that** you might fail*

- Pronouns:
  - shorthand referring to noun phrase etc.
  - Personal pronoun: *you, I, he, she, it*
  - Posessive pronoun: *your, mine, his, her, its, one's*
  - Wh-pronouns: *what, whom, whoever, why*

Part-of-Speech-Tags

- Auxiliary verbs:
  - mark semantic features of verbs: *can, do, may, should, are, have*: whether action is completed, negated, necessary, possible, suggested, desired, ….
  - Copula *be* : connects: *he* ***is*** *a duck*
  - Modal verbs: *can, must*

- Other classes:
  - Interjections *oh, hey, um, hmmm*
  - Negatives *no, not*
  - Politeness markers *please, thank you*
  - Greetings *hello, goodbye*
  - …

Penn Treebank POS 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*			

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

There/EX are/VBP 70/CD children/NNS **there/RB**

Preliminary/JJ findings/NNS were/VBD **reported/VBN** in/IN today/NN **’s/POS** New/NNP England/NNP Journal/NNP of/IN Medicine/NNP ./.

POS Labelled Corpora

- examples:
  - Brown corpus (1961, 10^6 words, different genre texts),
  - Wall Street Journal corpus (1989, 10^6 words),
  - Switchboard corpus (1991, $2*10^6$ words, telephone conversations)

- slight differences in using POS tags (e.g. in corpora)
  - e.g.
    -- Brown, WSJ: to/TO for both uses of to (preposition: *I like to dance;* infinitive: *too dangerous to swim*)
    -- Switchboard: Well/UH ,/, I/PRP ,/, I/PRP want/VBP to/TO go/VB to/IN a/DT restaurant/NN

POS Labelled Corpora

- POS tag sets: pragmatic decisions:
  - Penn 45 is a subset of larger POS tagsets, leaving off syntactic information recoverable from a parse tree, e.g. in Penn, the tag IN is used for subordinating conjunctions
    after/IN spending/VBG a/DT day/NN at/IN the/DT beach/NN
    as well as prepositions:
    after/IN sunrise/NN
  - Penn 45 assumes tokenization of multipart words:
    a/DT New/NNP York/NNP City/NNP firm/NN (New York City as one word)

POS Tagging

- After tokenization: POS tagging for each word: disambiguation task (*book a flight, read a book*)
  Not many words ambiguous but ambiguous words are among the most common tokens:

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%**)

- Most frequent POS tag (class) baseline: always predict the most frequent POS tag among the possible POS tags for an ambiguous word:
  on WSJ: accuracy: ≈0.92 ←→ state of the art: accuracy: ≈0.97

HMM for POS Tagging

- States: tags; observations: words
- training on labelled data: MLE by counting for A and B separately (No Baum Welch necessary):

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

- POS-Tagging via Viterbi algorithm: find:

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) = \underset{t_1^n}{\operatorname{argmax}} P(w_1^n|t_1^n)P(t_1^n)$$

- First oder Markov assumptions for A and B:

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

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

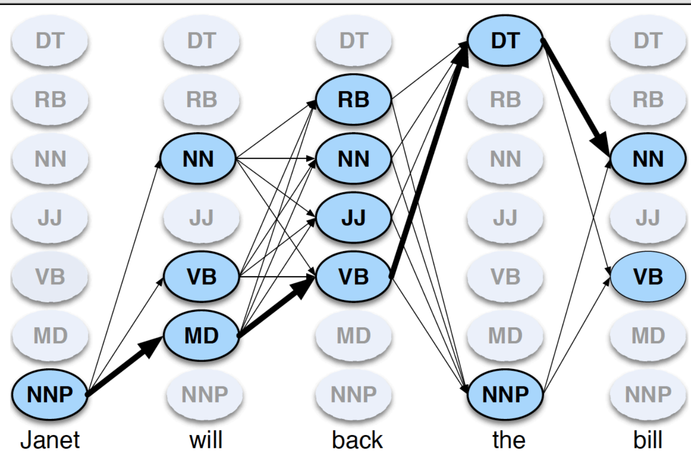
$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i|t_i)}^{\text{emission}} \overbrace{P(t_i|t_{i-1})}^{\text{transition}}$$

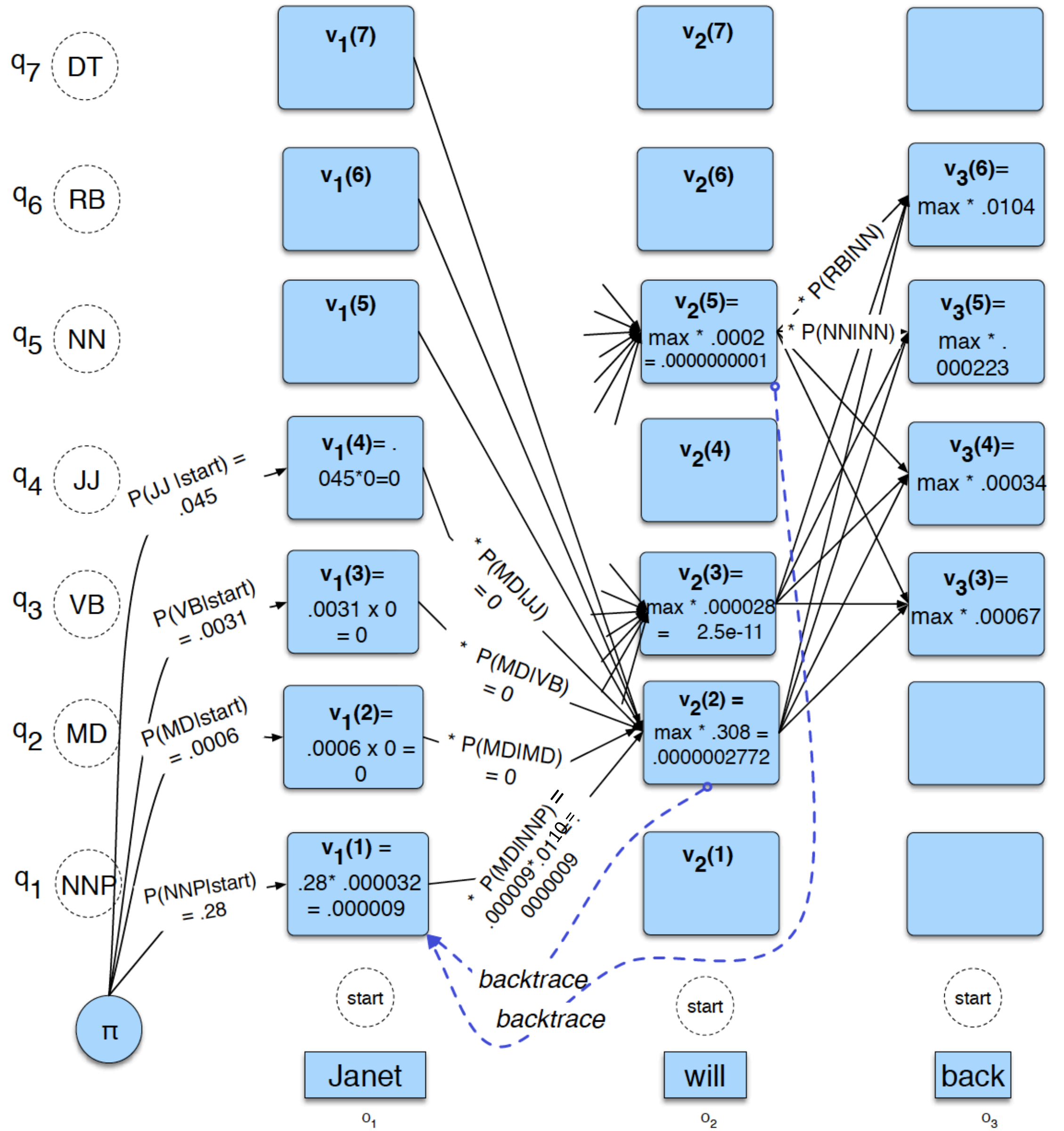
example: *Janet will back the bill* →
true POS tags:
Janet/NNP will/MD back/VB the/DT bill/NN

$$v_t(j) = \max_{q_{1:t-1}} P(q_{1:t-1}, q_t = j, o_{1:t} | \lambda)$$
$$= \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$$

	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

	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	0.000097	0
NN	0	0.000200	0.000223	0.000006	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0





Extending HMM to Trigrams

- switch to 2nd Markov order for transition model (we then have to add special "end of sentence markers" for t_{-1} , t_0 , t_{n+1}):

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \left[\prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1}, t_{i-2}) \right] P(t_{n+1} | t_n)$$

- problem (as in language modelling chapter): sparse counts for trigrams of tags. solution: back off to bigram or unigram + interpolate

$$\begin{aligned} \text{Trigrams} \quad \hat{P}(t_i | t_{i-1}, t_{i-2}) &= \frac{C(t_{i-2}, t_{i-1}, t_i)}{C(t_{i-2}, t_{i-1})} \\ \text{Bigrams} \quad \hat{P}(t_i | t_{i-1}) &= \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} \\ \text{Unigrams} \quad \hat{P}(t_i) &= \frac{C(t_i)}{N} \end{aligned}$$

$$P(t_i | t_{i-1} t_{i-2}) = \lambda_3 \hat{P}(t_i | t_{i-1} t_{i-2}) + \lambda_2 \hat{P}(t_i | t_{i-1}) + \lambda_1 \hat{P}(t_i)$$

Determine $\lambda_1 \lambda_2 \lambda_3$

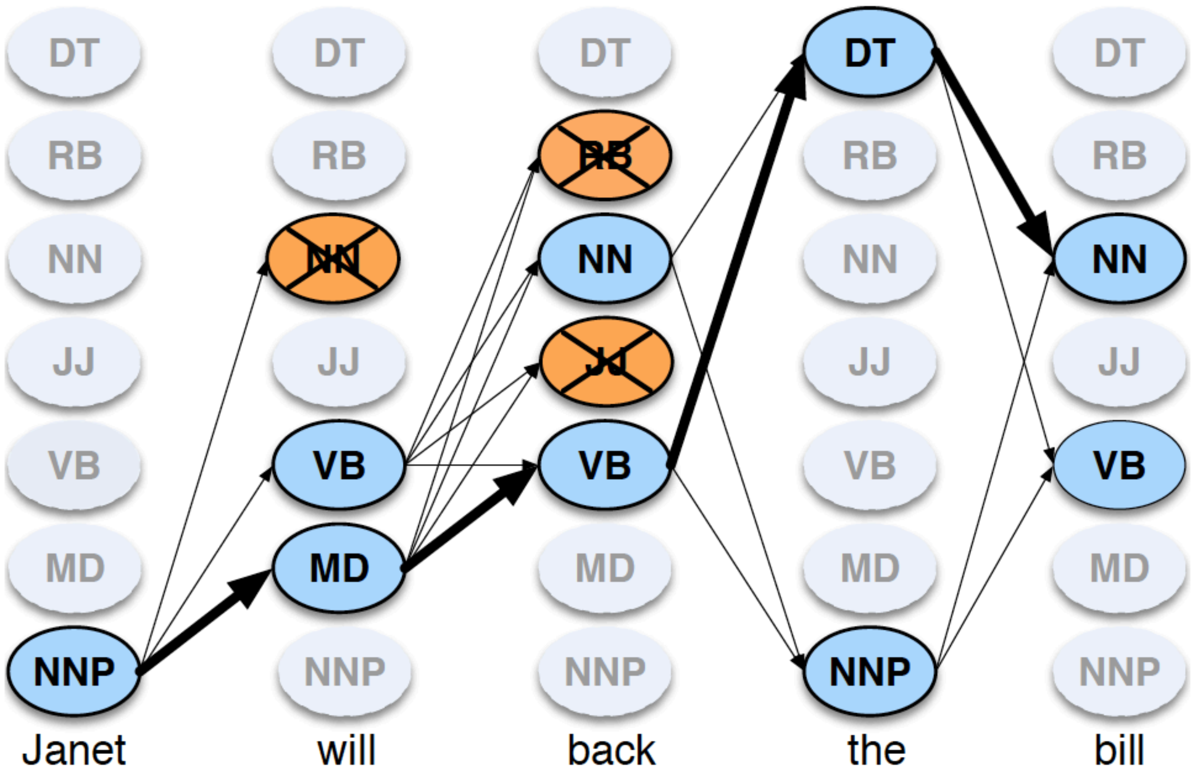
- successively delete each trigram from the training corpus and choose the λ s so as to maximize the likelihood of the rest of the corpus

function DELETED-INTERPOLATION(*corpus*) **returns** $\lambda_1, \lambda_2, \lambda_3$

$\lambda_1 \leftarrow 0$
$\lambda_2 \leftarrow 0$
$\lambda_3 \leftarrow 0$
foreach trigram t_1, t_2, t_3 with $C(t_1, t_2, t_3) > 0$
  depending on the maximum of the following three values
    case $\frac{C(t_1,t_2,t_3)-1}{C(t_1,t_2)-1}$: increment λ_3 by $C(t_1, t_2, t_3)$
    case $\frac{C(t_2,t_3)-1}{C(t_2)-1}$: increment λ_2 by $C(t_1, t_2, t_3)$
    case $\frac{C(t_3)-1}{N-1}$: increment λ_1 by $C(t_1, t_2, t_3)$
  end
end
normalize $\lambda_1, \lambda_2, \lambda_3$
return $\lambda_1, \lambda_2, \lambda_3$

Beam Search

- no of states N large → Viterbi ($O(N^2T)$) inefficient →
- instead of keeping all N=45 possibilities at each column, just concentrate on the β most probable ones (prune the possible hidden sequence tree);
- β : beam width



Dealing with Unknown Words

- Unseen words (no POS-tags for these occur in corpus): cannot estimate emission probabilities → use morphological information:

- examples suffixes: -s indicates plural noun, -able indicates adjective etc. → consider suffixes of unknown word w of length i with characters $l_{n-i+1}, \ldots, l_n$

$$P(t_i|w) \approx P(t_i|l_{n-i+1} \ldots l_n)$$

use Bayes theorem to get the required emission probability $P(w|t_i)$

- example capitalization: introduce boolean capitalization feature (effectively doubling tag set):

$$P(t_i|t_{i-1}, t_{i-2}) \rightarrow P(t_i, c_i|t_{i-1}, c_{i-1}, t_{i-2}, c_{i-2})$$

use Bayes theorem to get the required probabilities $P(c|t_i \ldots)$

Maximum Entropy Markov Models (MEMM) for POS Tagging

- switch to discriminative models (multinomial (softmax) logistic regression (maximum entropy models))

HMM (generative):

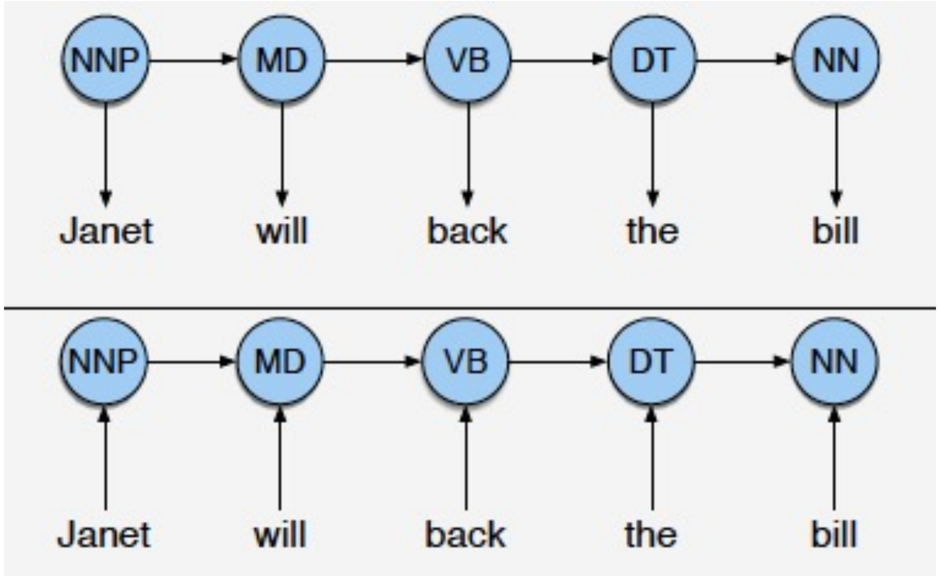
$$\begin{aligned}\hat{T} &= \underset{T}{\operatorname{argmax}} P(T|W) \\ &= \underset{T}{\operatorname{argmax}} P(W|T)P(T) \\ &= \underset{T}{\operatorname{argmax}} \prod_i P(w_i|t_i) \prod_i P(t_i|t_{i-1})\end{aligned}$$

MEMM (discriminative):

$$\begin{aligned}\hat{T} &= \underset{T}{\operatorname{argmax}} P(T|W) \\ &= \underset{T}{\operatorname{argmax}} \prod_i P(t_i|w_i,t_{i-1})\end{aligned}$$

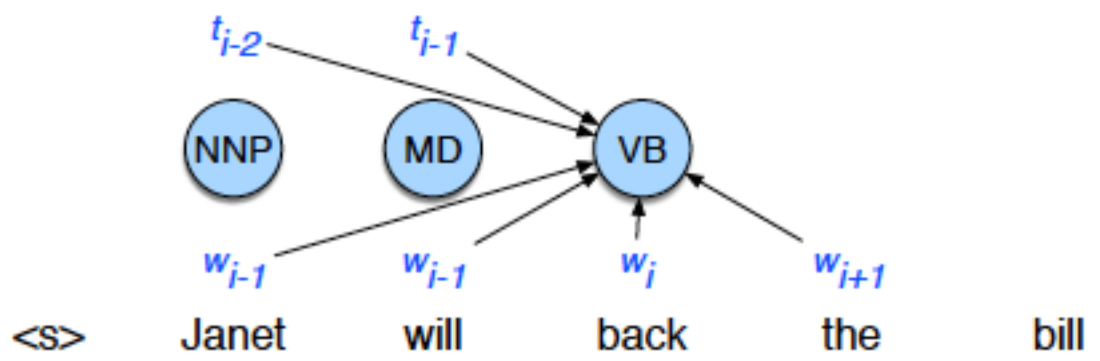
HMM:

MEMM:



Maximum Entropy Markov Models (MEMM) for POS Tagging

- advantage: much easier to incorporate even more features into condition-side (for HMM incorporating more features (like suffixes, capitalization) requires increasingly complex terms from Bayes theorem application)



- idea: use a lot of (often binary) features; use class(==tag)-dependent features & feature templates

Feature Templates

- example feature templates:

$$\langle t_i, w_{i-2} \rangle, \langle t_i, w_{i-1} \rangle, \langle t_i, w_i \rangle, \langle t_i, w_{i+1} \rangle, \langle t_i, w_{i+2} \rangle$$
$$\langle t_i, t_{i-1} \rangle, \langle t_i, t_{i-2}, t_{i-1} \rangle,$$
$$\langle t_i, t_{i-1}, w_i \rangle, \langle t_i, w_{i-1}, w_i \rangle \langle t_i, w_i, w_{i+1} \rangle,$$

- example : for Janet/NNP will/MD back/VB the/DT bill/NN and *w_i=back:*

$$t_i = \text{VB and } w_{i-2} = \text{Janet}$$
$$t_i = \text{VB and } w_{i-1} = \text{will}$$
$$t_i = \text{VB and } w_i = \text{back}$$
$$t_i = \text{VB and } w_{i+1} = \text{the}$$
$$t_i = \text{VB and } w_{i+2} = \text{bill}$$
$$t_i = \text{VB and } t_{i-1} = \text{MD}$$
$$t_i = \text{VB and } t_{i-1} = \text{MD and } t_{i-2} = \text{NNP}$$
$$t_i = \text{VB and } w_i = \text{back and } w_{i+1} = \text{the}$$

Feature Templates

- other possible features (especially for unknown words) :

w_i contains a particular prefix (from all prefixes of length ≤ 4)
w_i contains a particular suffix (from all suffixes of length ≤ 4)
w_i contains a number
w_i contains an upper-case letter
w_i contains a hyphen
w_i is all upper case
w_i's word shape
w_i's short word shape
w_i is upper case and has a digit and a dash (like *CFC-12*)
w_i is upper case and followed within 3 words by Co., Inc., etc.

Feature Templates

- furthermore: word shape features: x: letter; X: uppercase letter; d: number; punctuation

- exampe: *well-dressed*

$$\text{prefix}(w_i) = \texttt{w}$$
$$\text{prefix}(w_i) = \texttt{we}$$
$$\text{prefix}(w_i) = \texttt{wel}$$
$$\text{prefix}(w_i) = \texttt{well}$$
$$\text{suffix}(w_i) = \texttt{ssed}$$
$$\text{suffix}(w_i) = \texttt{sed}$$
$$\text{suffix}(w_i) = \texttt{ed}$$
$$\text{suffix}(w_i) = \texttt{d}$$
$$\text{has-hyphen}(w_i)$$
$$\text{word-shape}(w_i) = \texttt{xxxx-xxxxxxx}$$
$$\text{short-word-shape}(w_i) = \texttt{x-x}$$

Maximum Entropy Markov Models (MEMM) for POS Tagging

Given large set of features computed from the word w_i the l previous words w_{i-l}^{i+l} and the k previous tags t_{i-k}^{i-1}

$$
\begin{aligned}
\hat{T} &= \operatorname*{argmax}_T P(T|W) \\
&= \operatorname*{argmax}_T \prod_i P(t_i | w_{i-l}^{i+l}, t_{i-k}^{i-1}) \\
&= \operatorname*{argmax}_T \prod_i \frac{\exp\left(\sum_j \theta_j f_j(t_i, w_{i-l}^{i+l}, t_{i-k}^{i-1})\right)}{\sum_{t' \in \text{tagset}} \exp\left(\sum_j \theta_j f_j(t', w_{i-l}^{i+l}, t_{i-k}^{i-1})\right)}
\end{aligned}
$$

θ_j: weights of multiclass logistic regression model

Using MEMMs for POS-Tagging

- simple approach: use log. Regr. classifier for each word separately (greedy):

function GREEDY MEMM DECODING(words W, model P) **returns** tag sequence T

for $i = 1$ **to** $length(W)$

$$\hat{t}_i = \underset{t' \in T}{\operatorname{argmax}} P(t' \mid w_{i-l}^{i+l}, t_{i-k}^{i-1})$$

disadvantage: makes hard decision before moving on, can't use future evidence as Viterbi can (←→ backtrace) → don't use it!

- significantly better: incorporate decisions on other words: use Viterbi: (assume for simplicity that MEMM uses only Markov order 1 features):

HMM Viterbi:

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

$$= \max_{i=1}^{N} v_{t-1}(i)\, P(s_j|s_i)\, P(o_t|s_j)$$

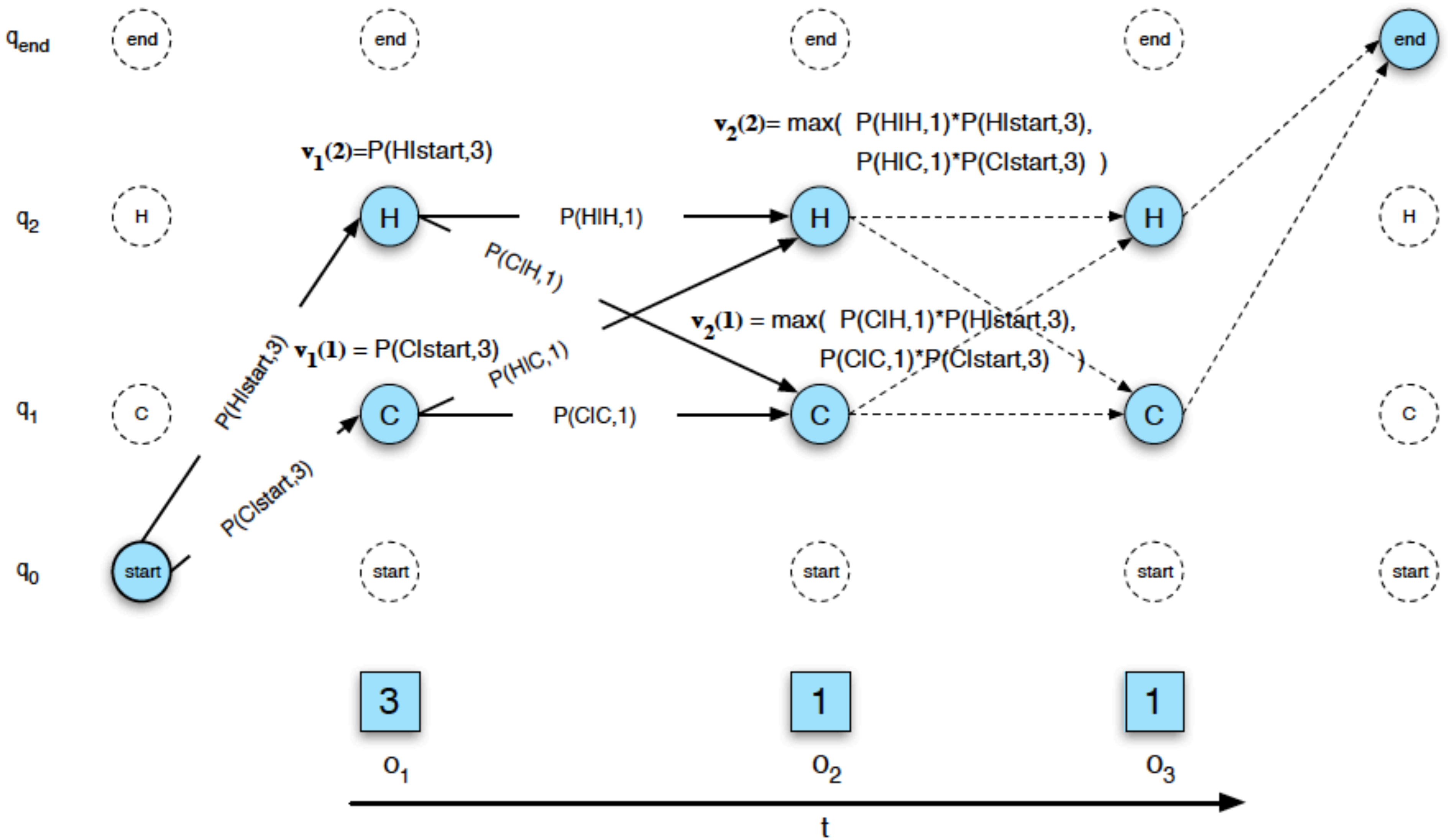
MEMM Viterbi:

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, P(s_j|s_i, o_t)$$

MEMM Viterbi

use ice cream eating example
from chapter 9

$$v_t(j) \;=\; \max_{i=1}^{N} \; v_{t-1}(i)\, P(s_j|s_i,o_t)$$



Disadvantages of MEMM

- HMM and MEMM's computations (the recursion) are uni-directional (left to right). possibly better: incorporate future tags (bi-directionality), e.g. by switching to Conditional Random Fields (CRFs)

- Other way to incorporate bi-directionality implicitly: do multiple passes with MEMM. on second and later passes: incorporate features using future tags (computed / improved upon by previous passes

Disadvantages of MEMM

- General weakness of MEMM: label bias / observational bias:
  example: will/NN to/TO fight/VB.
  - fact: *to* is often preceded by NN but less often by modals MD.
  - fact: $P(t_{will}|$ <s>$)$ is generally large for t_{will}=MD (e.g. <s>*Will you follow me?*</s>).
  - fact: $P(\text{TO}|\textit{to}, t_{will}) \approx 1$ regardless of t_{will}
  - → $P(\text{TO}|\textit{to}, t_{will}) \approx 1$ explains away the TO for *to*, disregarding the importance of the previous NN.
  - → because $P(t_{will}|$ <s>$)$ is large for t_{will}=MD and because the transition $P(t_{will} = NN, t_{to} = TO)$ is practically ignored due to the explaining away of to by $P(\text{TO}|\textit{to}, t_{will}) \approx 1$ regardless of t_{will} , t_{will} will be wrongfully assigned MD instead of the correct NN

Bibliography

(1) Dan Jurafsky and James Martin: Speech and Language Processing (3rd ed. draft, version Oct2019); Online: https://web.stanford.edu/~jurafsky/slp3/ (URL, Oct 2019) (this slide-set is especially based on chapter 8)

Recommendations for Studying

- minimal approach:

  work with the slides and understand their contents! Think beyond instead of merely memorizing the contents

- standard approach:

  minimal approach + read the corresponding pages in Jurafsky [1]

- interested students

  == standard approach