POS tagging

Natural Language Processing: Lecture 5

5.10.2017

Kairit Sirts

Plan

- What are POS tags?
- POS tagging task as a disambiguation task
- Approaches to POS tagging:
  - Hidden Markov models and Viterbi decoding
  - Log-linear models
  - Neural networks

POS tags

- Part-of-speech tags, syntactic categories, word classes
- POS tags give information about the word and its neighbors
- Useful for many other NLP tasks: information extraction, syntactic parsing, information retrieval, summarization

Janet	will	back	the	bill
Proper noun	Modal verb	Verb	Article	Noun

Main POS categories

- Nouns
  - Common nouns – things (chair), events (lecture) , abstractions (justice) , verb-like terms (swimming) etc
  - Proper nouns – proper names of people (John), countries (Estonia), organizations (University of Tartu) etc
- Verbs – words referring to actions and processes (to draw, to ponder)
- Adjectives – words describing properties or qualities (black, young)
- Adverbs – words modifying (mostly) verbs

Unfortunately, John walked home extremely slowly yesterday

Open and closed class words

Open class words

- Nouns
- Verbs
- Adjectives
- Adverbs

Closed class words

- Prepositions: on, under, over
- Determiners: a, an, the
- Pronouns: she, who, I, others
- Conjunctions: and, but, or, as
- Auxiliary verbs: can, may, are
- Particles: up, down, on, off
- Numerals: one, two, first

English POS tags – Penn Treebank tagset

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*			

Speech and Language Processing: https://web.stanford.edu/~jurafsky/slp3/10.pdf, page 5

Estonian POS tags – used in estnltk

POS tag	Description	Example	POS tag	Description	Example
A	Adjective	kallis	N	Cardinal number	kaks
C	Adj., comparative	laiem	O	Ordinal number	teine
D	Adverb	kõrvuti	P	Pronoun	see
G	Genitive attribute	balti	S	Common noun	asi
H	Proper name	Edgar	U	Adj., superlative	pikim
I	Interjection	tere	V	Verb	lugema
J	Conjunction	ja	X	Verb complements	plehku
K	Adposition	kaudu	Y	Abbreviation	USA
			Z	Punctuation	-, /, …

Morphological tags

	Nouns	Verbs
English	2 numbers (singular, plural)	6 different forms
	2 cases (nominative, genitive)	
Estonian	2 numbers	Tens of different forms
	14 cases	

http://www.filosoft.ee/html_morf_et/morfoutinfo.html

POS tags in Multext-East corpus

Language	Number of tags
English	104
Polish	588
Hungarian	429
Czech	573
Serbian	456
Farsi	207
Bulgarian	104
Slovak	581
Slovene	610
Estonian	316

Kairit Sirts. Non-parametric Bayesian Models for Computational Morphology, page 138.

Universal POS tags

POS tag	Description
VERB	Verbs (all tenses and modes)
NOUN	Nouns (common and proper)
PRON	Pronouns
ADJ	Adjectives
ADV	Adverbs
ADP	Adpositions (prepositions and postpositions)
CONJ	Conjunctions
DET	Determiners
NUM	Cardinal numbers
PRT	Particles or other function words
X	Other: foreign words, typos, abbreviations
.	Punctuation

Petrov et al., 2011. A Universal Part-of-Speech Tagset.

POS tagging

Janet	will	back	the	bill
NNP	MD	VB	DT	NN
NOUN	VERB	VERB	DET	NOUN

Teise	koha	sai	seekord	Jänes
O+sg_g	S+sg_g	V+s	D	H+sg_n
NUM	NOUN	VERB	ADV	NOUN

POS tagging as a disambiguation task

Teise	koha	sai	seekord	Jänes
O+adt	S+sg_g	V+s	D	S+sg_n
O+sg_g	S+sg_n	S+sg_n		H+sg_n
P+adt	S+sg_p			
P+sg_g	V+o			

POS tagging as a disambiguation task

Teise	koha	sai	seekord	Jänes
O+adt	**S+sg_g**	**V+s**	**D**	S+sg_n
O+sg_g	S+sg_n	S+sg_n		**H+sg_n**
P+adt	S+sg_p			
P+sg_g	V+o			

Tag ambiguity in English

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%**)

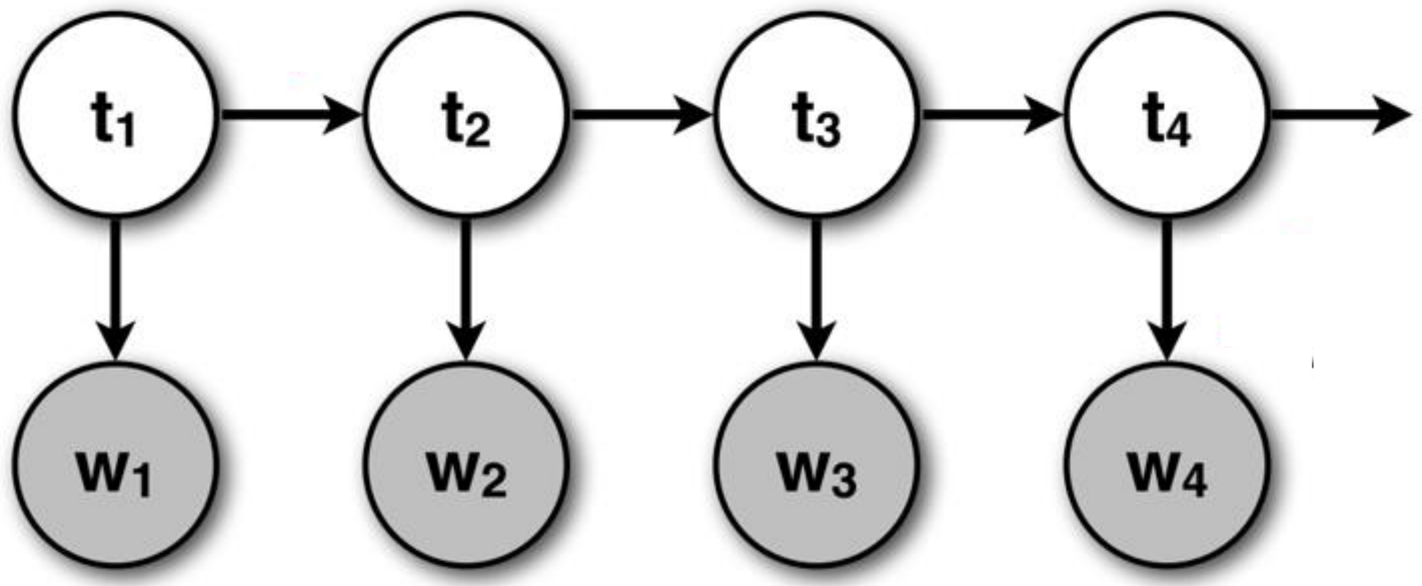
Figure 10.2 The amount of tag ambiguity for word types in the Brown and WSJ corpora, from the Treebank-3 (45-tag) tagging. These statistics include punctuation as words, and assume words are kept in their original case.

https://web.stanford.edu/~jurafsky/slp3/10.pdf

POS tagging approaches

1. Hidden Markov Models
2. Log-linear models
3. Neural networks

Hidden Markov Model



http://www.cs.virginia.edu/~hw5x/Course/CS6501-Text-Mining/_site/mps/mp3.html

Hidden Markov Model

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

$$= \arg\max_{t_1^n} \frac{P(w_1^n | t_1^n) P(t_1^n)}{P(w_1^n)}$$

$$= \arg\max_{t_1^n} P(w_1^n | t_1^n) P(t_1^n)$$

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

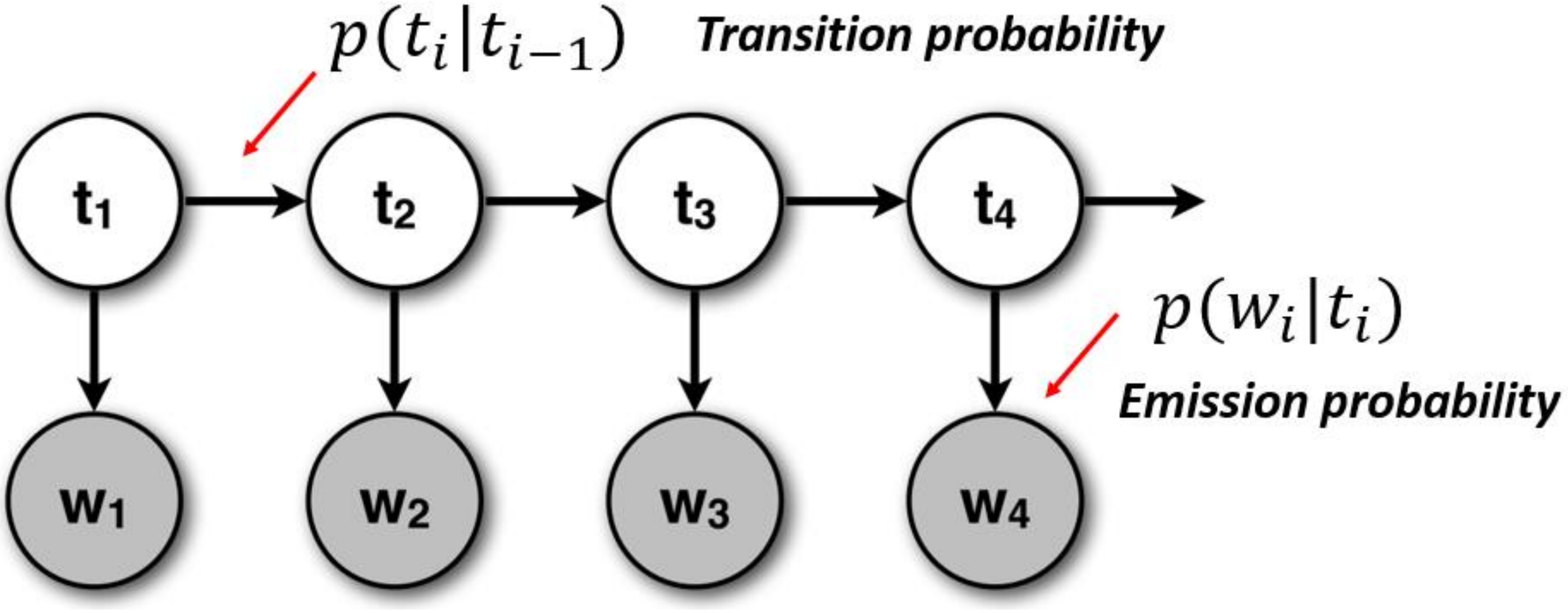
Application of the Bayes rule:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

Argmax does not depend on the denominator, thus can be omitted

Factor the probabilities according to the dependencies in the graphical model

Hidden Markov Model



Parameter estimation

Transition probabilities

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

$$P(VB|MD) = \frac{C(MD, VB)}{C(MD)}$$

$$= \frac{10471}{13124} = 0.80$$

Emission probabilities

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

$$P(will|MD) = \frac{C(MD, will)}{C(MD)}$$

$$= \frac{4046}{13124} = 0.31$$

Viterbi decoding

- The goal is to find the most likely sequence of hidden tags
- Viterbi decoding is a dynamic programming algorithm

In computer science, mathematics, management science, economics and bioinformatics, **dynamic programming** (also known as **dynamic optimization**) is a method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions. The next time the same subproblem occurs, instead of recomputing its solution, one simply looks up the previously computed solution, thereby saving computation time at the expense of a (hopefully) modest expenditure in storage space.

https://en.wikipedia.org/wiki/Dynamic_programming

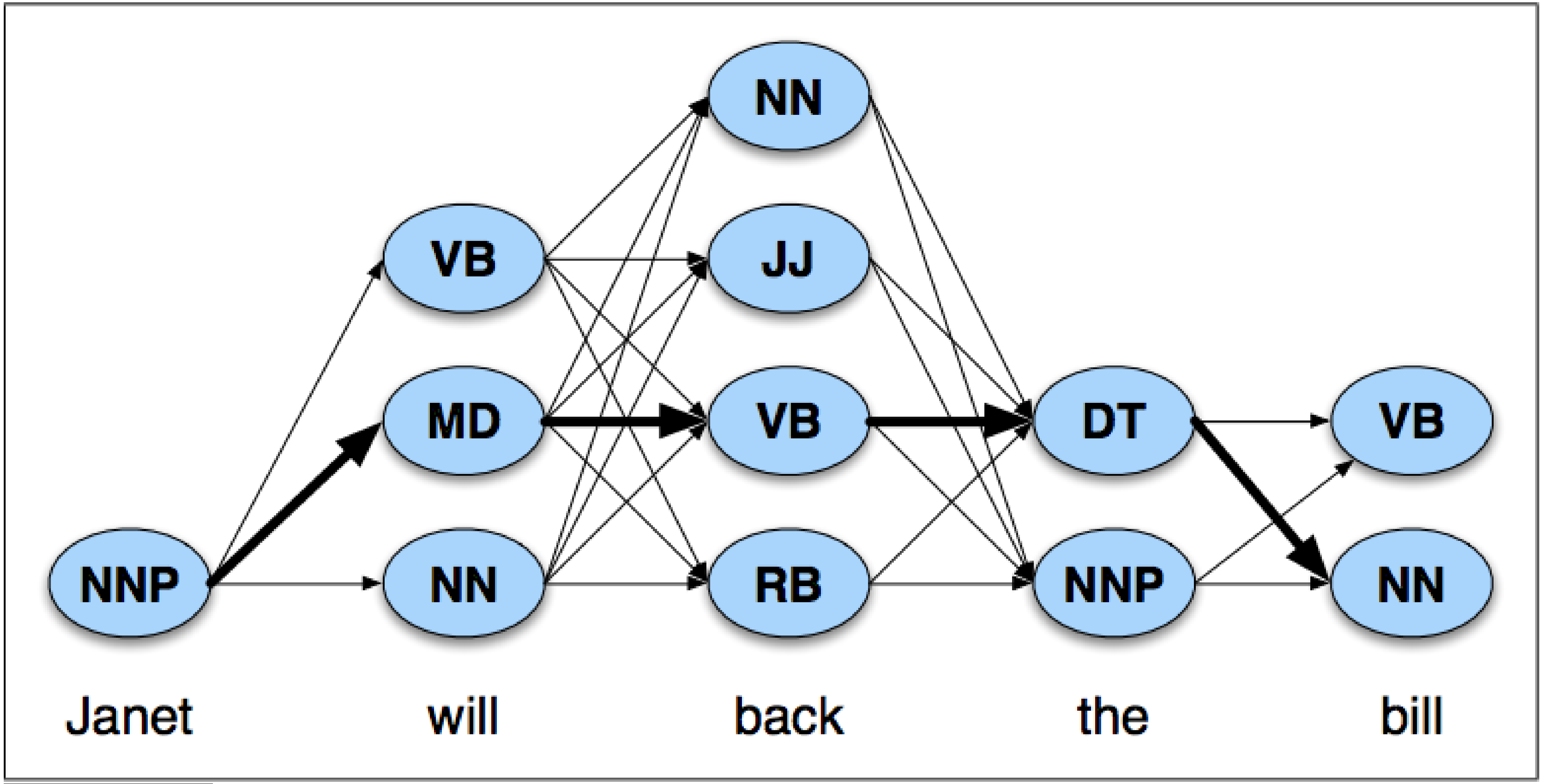


Figure 10.7 A schematic of the tagging task for the sample sentence, showing the ambiguities for each word and the correct tag sequence as the highlighted path through the hidden states.

https://web.stanford.edu/~jurafsky/slp3/10.pdf

	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

Figure 10.5 The A transition probabilities $P(t_i|t_{i-1})$ computed from the WSJ corpus without smoothing. Rows are labeled with the conditioning event; thus $P(VB|MD)$ is 0.7968.

Transition probabilities

	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

Figure 10.6 Observation likelihoods B computed from the WSJ corpus without smoothing.

Emission probabilities

https://web.stanford.edu/~jurafsky/slp3/10.pdf

Viterbi algorithm

- Create a table V with N+2 rows and T columns:
  - N – the number of states/tags
  - T – the length of the sequence/sentence
- Initialise the first column
  - For each tag t in the tagset compute:

$$V[t,1] = P(t|start)P(w_1|t)$$

- For each column j = 2 to T in the table V:
  - For each tag t in the tagset compute:

$$V[t,j] = \max_{t'} V[t',j-1]P(t|t')P(w_j|t)$$

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009				
MD	0				
VB	0				
JJ	0				
NN	0				
RB	0				
DT	0				

	NNP
$<s>$	0.2767
NNP	0.3777
MD	0.0008
VB	0.0322
JJ	0.0366
NN	0.0096
RB	0.0068
DT	0.1147

	Janet
NNP	0.000032
MD	0
VB	0
JJ	0
NN	0
RB	0
DT	0

$$V[t,1] = P(t|start)P(w_1|t)$$

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0			
MD	0	0.000009*0.011* 0.308431=3E-8			
VB	0	0.000009*0.0009* 0.000028=2.3E-13			
JJ	0	0			
NN	0	0.000009*0.0584* 0.0002=1.1E-10			
RB	0	0			
DT	0	0			

	Janet	will
NNP	0.000032	0
MD	0	0.308431
VB	0	0.000028
JJ	0	0
NN	0	0.000200
RB	0	0
DT	0	0

	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

$$V[t,2] = \max_{t'} V[t',1]P(t|t')P(w_2|t)$$

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0	0		
MD	0	0.000009*0.011* 0.308431=3E-8	0		
VB	0	0.000009*0.0009* 0.000028=2.3E-13			
JJ	0	0			
NN	0	0.000009*0.0584* 0.0002=1.1E-10			
RB	0	0			
DT	0	0	0		

	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

$$V[t,3] = \max_{t'} V[t',2] P(t|t') P(w_3|t)$$

back

	VB	JJ	NN	RB
MD 3E-8	***0.7968=2.4E-8**	***0.0005=1.5E-11**	*0.0008=2.4E-11	***0.009=2.7E-10**
VB 2.3E-13	*0.005=1.5E-15	*0.0837=1.9E-14	*0.0615=1.4E-14	*0.0514=1.2E-14
NN 1.1E-10	*0.0014=1.5E-13	*0.0086=9.5E-13	***0.1216=1.3E-11**	*0.0177=1.9E-12

	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

$$\max_{t'} V[t', 2] P(t|t')$$

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0	0		
MD	0	0.000009*0.011* 0.308431=3E-8	0		
VB	0	0.000009*0.0009* 0.000028=2.3E-13	3E-8*0.7968* 0.000672=1.6E-11		
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15		
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15		
RB	0	0	3E-8*0.009* 0.010446=2.8E-12		

	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

$$V[t,3] = \max_{t'} V[t',2]P(t|t')P(w_3|t)$$

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0	0		
MD	0	0.000009*0.011* 0.308431=3E-8	0	0	
VB	0	0.000009*0.0009* 0.000028=2.3E-13	3E-8*0.7968* 0.000672=1.6E-11	0	
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15		
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15		
RB	0	0	3E-8*0.009* 0.010446=2.8E-12	0	

	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

$$V[t,4] = \max_{t'} V[t',3]P(t|t')P(w_4|t)$$

the

	NNP	JJ	NN	DT
VB 1.6E-11	***0.0322=5.2E-13**	***0.0837=1.3E-12**	***0.0615=9.8E-13**	***0.2231=3.6E-12**
JJ 5.1E-15	*0.0366=1.9E-16	*0.0733=3.7E-16	*0.4509=2.3E-15	*0.0036=1.8E-17
NN 3E-15	*0.0096=2.9E-17	*0.0086=2.6E-17	*0.1216=3.6E-16	*0.0068=2E-17
RB 2.8E-12	*0.0068=1.9E-14	*0.1012=2.8E-13	*0.0120=3.4E-14	*0.0479=1.3E-13

	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

$$\max_{t'} V[t', 3] P(t|t')$$

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0	0	1.6E-11*0.0322* 0.000048=2.5E-17	
MD	0	0.000009*0.011* 0.308431=3E-8	0	0	
VB	0	0.000009*0.0009* 0.000028=2.3E-13	3E-8*0.7968* 0.000672=1.6E-11	0	
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15	1.6E-11*0.0837* 0.000097=1.3E-16	
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15	1.6E-11*0.0615* 0.000006=5.9E-18	
RB	0	0	3E-8*0.009* 0.010446=2.8E-12	0	
				1.6E-11*0.2231* 0.506099=1.8E-12	

	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

$$V[t,4] = \max_{t'} V[t',3]P(t|t')P(w_4|t)$$

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0	0	1.6E-11*0.0322* 0.000048=2.5E-17	0
MD	0	0.000009*0.011* 0.308431=3E-8	0	0	0
VB	0	0.000009*0.0009* 0.000028=2.3E-13	3E-8*0.7968* 0.000672=1.6E-11	0	
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15	1.6E-11*0.0837* 0.000097=1.3E-16	0
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15	1.6E-11*0.0615* 0.000006=5.9E-18	
RB	0	0	3E-8*0.009* 0.010446=2.8E-12	0	0
				1.6E-11*0.2231* 0.506099=1.8E-12	0

	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

$$V[t,5] = \max_{t'} V[t',4]P(t|t')P(w_5|t)$$

bill

	VB	NN
NNP 2.5E-17	*0.0009=2.2E-20	*0.0584=1.5E-18
JJ 1.3E-16	*0.0001=1.3E-20	*0.4509=5.9E-17
NN 5.9E-18	*0.0014=8.3E-21	*0.1216=7.2E-19
DT 1.8E-12	***0.0002=3.6E-16**	***0.4744=8.5E-13**

	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

$$\max_{t'} V[t', 4]P(t|t').$$

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0	0	1.6E-11*0.0322* 0.000048=2.5E-17	0
MD	0	0.000009*0.011* 0.308431=3E-8	0	0	0
VB	0	0.000009*0.0009* 0.000028=2.3E-13	3E-8*0.7968* 0.000672=1.6E-11	0	1.8E-12*0.0002* 0.000028=1E-20
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15	1.6E-11*0.0837* 0.000097=1.3E-16	0
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15	1.6E-11*0.0615* 0.000006=5.9E-18	1.8E-12*0.4744* 0.002337=2E-15
RB	0	0	3E-8*0.009* 0.010446=2.8E-12	0	0
DT	0	0	0	1.6E-11*0.2231* 0.506099=1.8E-12	0

	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

$$V[t,5] = \max_{t'} V[t',4]P(t|t')P(w_5|t)$$

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0	0	1.6E-11*0.0322* 0.000048=2.5E-17	0
MD	0	0.000009*0.011* 0.308431=3E-8	0	0	0
VB	0	0.000009*0.0009* 0.000028=2.3E-13	3E-8*0.7968* 0.000672=1.6E-11	0	1.8E-12*0.0002* 0.000028=1E-20
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15	1.6E-11*0.0837* 0.000097=1.3E-16	0
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15	1.6E-11*0.0615* 0.000006=5.9E-18	**1.8E-12*0.4744* 0.002337=2E-15**
RB	0	0	3E-8*0.009* 0.010446=2.8E-12	0	0
DT	0	0	0	1.6E-11*0.2231* 0.506099=1.8E-12	0

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0	0	1.6E-11*0.0322* 0.000048=2.5E-17	0
MD	0	0.000009*0.011* 0.308431=3E-8	0	0	0
VB	0	0.000009*0.0009* 0.000028=2.3E-13	3E-8*0.7968* 0.000672=1.6E-11	0	1.8E-12*0.0002* 0.000028=1E-20
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15	1.6E-11*0.0837* 0.000097=1.3E-16	0
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15	1.6E-11*0.0615* 0.000006=5.9E-18	**1.8E-12*0.4744* 0.002337=2E-15**
RB	0	0	3E-8*0.009* 0.010446=2.8E-12	0	0
DT	0	0	0	**1.6E-11*0.2231* 0.506099=1.8E-12**	0

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0	0	1.6E-11*0.0322* 0.000048=2.5E-17	0
MD	0	0.000009*0.011* 0.308431=3E-8	0	0	0
VB	0	0.000009*0.0009* 0.000028=2.3E-13	**3E-8*0.7968* 0.000672=1.6E-11**	0	1.8E-12*0.0002* 0.000028=1E-20
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15	1.6E-11*0.0837* 0.000097=1.3E-16	0
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15	1.6E-11*0.0615* 0.000006=5.9E-18	**1.8E-12*0.4744* 0.002337=2E-15**
RB	0	0	3E-8*0.009* 0.010446=2.8E-12	0	0
DT	0	0	0	**1.6E-11*0.2231* 0.506099=1.8E-12**	0

	Janet	will	back	the	bill
NNP	0.2767* 0.000032= 0.000009	0	0	1.6E-11*0.0322* 0.000048=2.5E-17	0
MD	0	**0.000009*0.011* 0.308431=3E-8**	0	0	0
VB	0	0.000009*0.0009* 0.000028=2.3E-13	**3E-8*0.7968* 0.000672=1.6E-11**	0	1.8E-12*0.0002* 0.000028=1E-20
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15	1.6E-11*0.0837* 0.000097=1.3E-16	0
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15	1.6E-11*0.0615* 0.000006=5.9E-18	**1.8E-12*0.4744* 0.002337=2E-15**
RB	0	0	3E-8*0.009* 0.010446=2.8E-12	0	0
DT	0	0	0	**1.6E-11*0.2231* 0.506099=1.8E-12**	0

	Janet	will	back	the	bill
NNP	**0.2767* 0.000032= 0.000009**	0	0	1.6E-11*0.0322* 0.000048=2.5E-17	0
MD	0	**0.000009*0.011* 0.308431=3E-8**	0	0	0
VB	0	0.000009*0.0009* 0.000028=2.3E-13	**3E-8*0.7968* 0.000672=1.6E-11**	0	1.8E-12*0.0002* 0.000028=1E-20
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15	1.6E-11*0.0837* 0.000097=1.3E-16	0
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15	1.6E-11*0.0615* 0.000006=5.9E-18	**1.8E-12*0.4744* 0.002337=2E-15**
RB	0	0	3E-8*0.009* 0.010446=2.8E-12	0	0
DT	0	0	0	**1.6E-11*0.2231* 0.506099=1.8E-12**	0

	NNP	MD	VB	DT	NN
	Janet	will	back	the	bill
NNP	**0.2767* 0.000032= 0.000009**	0	0	1.6E-11*0.0322* 0.000048=2.5E-17	0
MD	0	**0.000009*0.011* 0.308431=3E-8**	0	0	0
VB	0	0.000009*0.0009* 0.000028=2.3E-13	**3E-8*0.7968* 0.000672=1.6E-11**	0	1.8E-12*0.0002* 0.000028=1E-20
JJ	0	0	3E-8*0.0005* 0.00034=5.1E-15	1.6E-11*0.0837* 0.000097=1.3E-16	0
NN	0	0.000009*0.0584* 0.0002=1.1E-10	1.1E-10*0.1216* 0.000223=3E-15	1.6E-11*0.0615* 0.000006=5.9E-18	**1.8E-12*0.4744* 0.002337=2E-15**
RB	0	0	3E-8*0.009* 0.010446=2.8E-12	0	0
DT	0	0	0	**1.6E-11*0.2231* 0.506099=1.8E-12**	0

POS tagging using HMM

- The so-called “classical” method
- Can use smoothing and interpolating similar to ngram language modeling to include more information
- Prefix and suffix probabilities for unknown words
- Expectation-maximization for unsupervised POS tagging
- Estonian morphological disambiguator (accessible in estnltk) uses HMM.

POS tagging with log-linear models

- Maximum entropy markov model
- Bidirectional MEMM – Stanford POS tagger
- Conditional Random Fields (CRF)

Maximum Entropy Markov Model

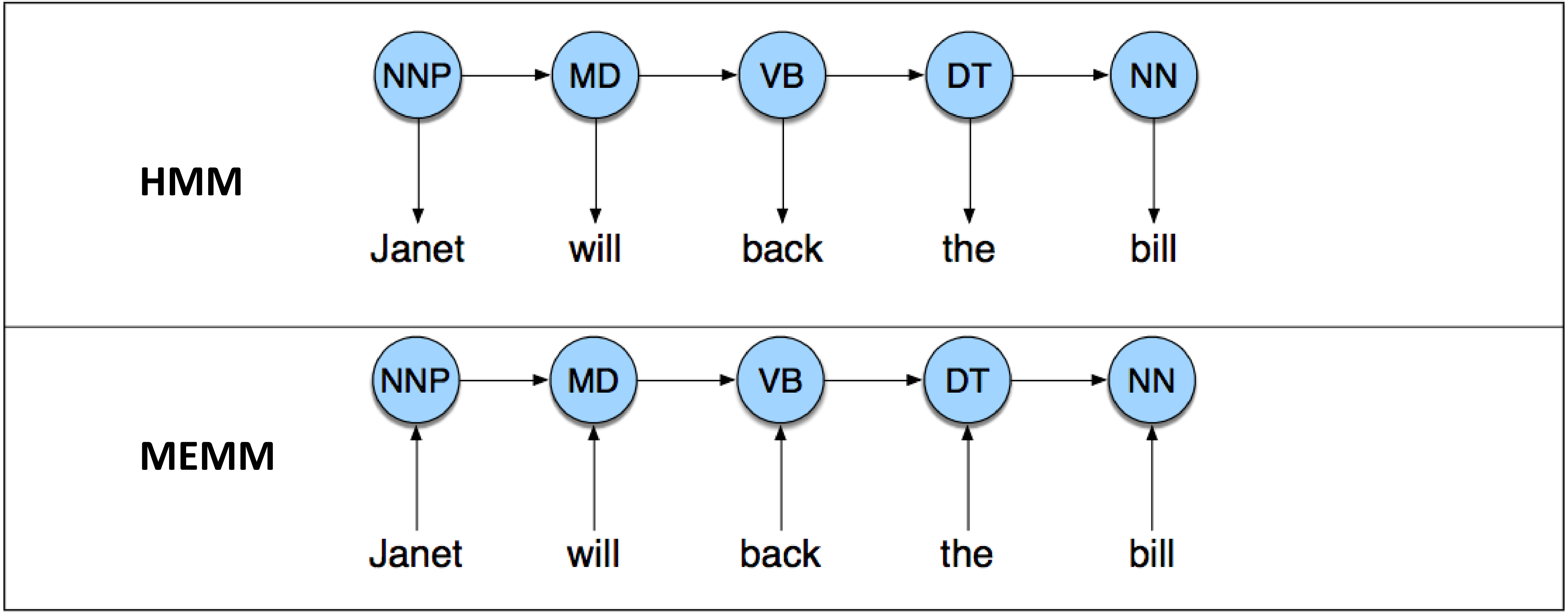
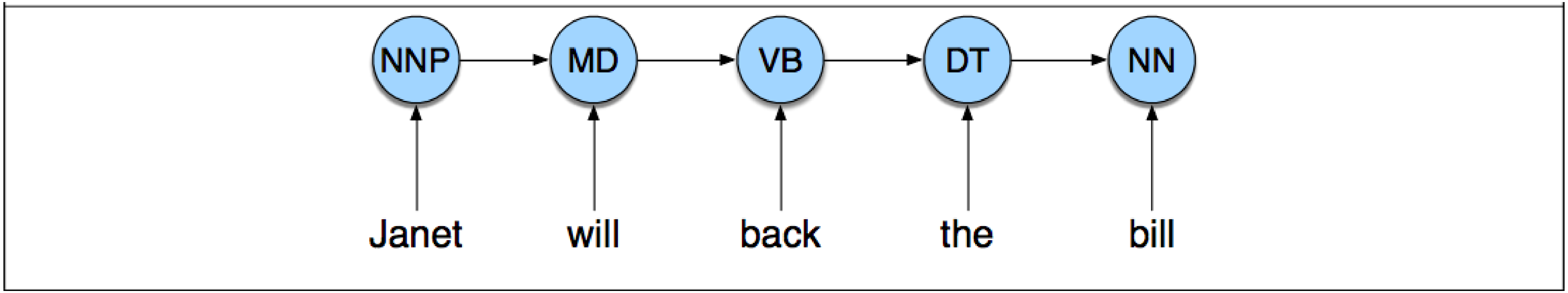


Figure 10.11: https://web.stanford.edu/~jurafsky/slp3/10.pdf

MEMM



$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

$$= \arg\max_{t_1^n} \prod_{i=1}^{n} P(t_i | t_{i-1}, w_i)$$

$$= \arg\max_{t_1^n} \prod_{i=1}^{n} P(t_i | t_1^{i-1}, w_1^n)$$

MEMM

- Each factorized component $P(t_i|t_1^{i-1}, w_1^n)$ is a local maximum entropy classifier
- The features can be extracted from the currently predicted tag, the tags predicted for the previous words and from the whole word sequence
- The cost function for one local model has a familiar log-linear form:

$$P(t_i|t_1^{i-1}, w_1^n) = \frac{\exp(v \cdot f(t_i, t_1^{i-1}, w_1^n))}{\sum_{t \in T} \exp(v \cdot f(t, t_1^{i-1}, w_1^n))}$$

Word and tag feature templates

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

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

https://web.stanford.edu/~jurafsky/slp3/10.pdf

Feature templates 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.

Features for the word **well-dressed**

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

https://web.stanford.edu/~jurafsky/slp3/10.pdf

Decoding in MEMM

Greedy decoding

- Proceed from left to right and make predictions

PRO:

- Very quick

CON:

- Local solutions, future cannot influence the present

Viterbi decoding

- Change the standard Viterbi by replacing the transition and emission probabilities with the log-linear probability

PRO:

- Future decisions can influence earlier decisions

CON:

- Slower (but not too slow)

Label bias problem

- Occurs because the models are locally normalised

will **to** **fight**

NN? TO
MD?

P(TO|to) =1

P(TO|NN) = large
P(TO|MD) = small

P(MD|START) > P(NN|START)

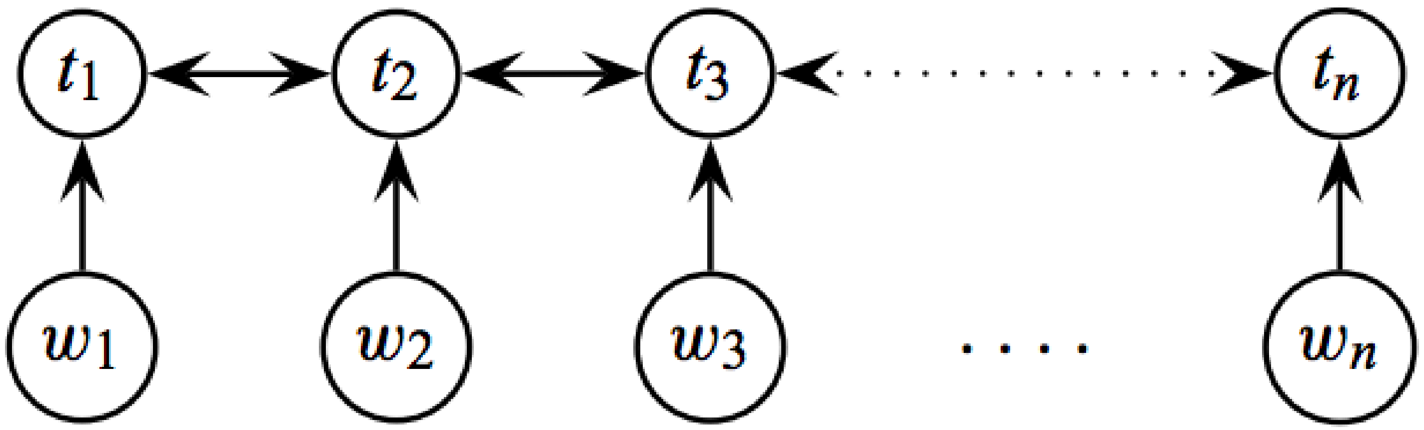
P(TO|to, t(will)) = P(TO|to) =1

P(TO|to, MD) = 1
P(TO|to, NN) = 1

Example from: Toutanova et al., 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network

Bidirectional MEMM

- Stanford POS tagger
  - Toutanova et al., 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network
  - Accuracy on English WSJ corpus 97.24%



POS tagging with neural models

- Feed-forward neural network
  - Collobert et al., 2011. Natural Language Processing (Almost) from Scratch
  - Andor et al., 2016. Globally Normalized Transition-Based Neural Networks
- Bidirectional LSTM
  - Ma and Hovy, 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF
  - Huang et al., 2015. Bidirectional LSTM-CRF Models for Sequence Tagging
  - Ling et al., 2015. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation

Collobert & Weston, 2011

- Senna word embeddings
- Feed-forward network
- Predicts a POS tag looking at a local window of words

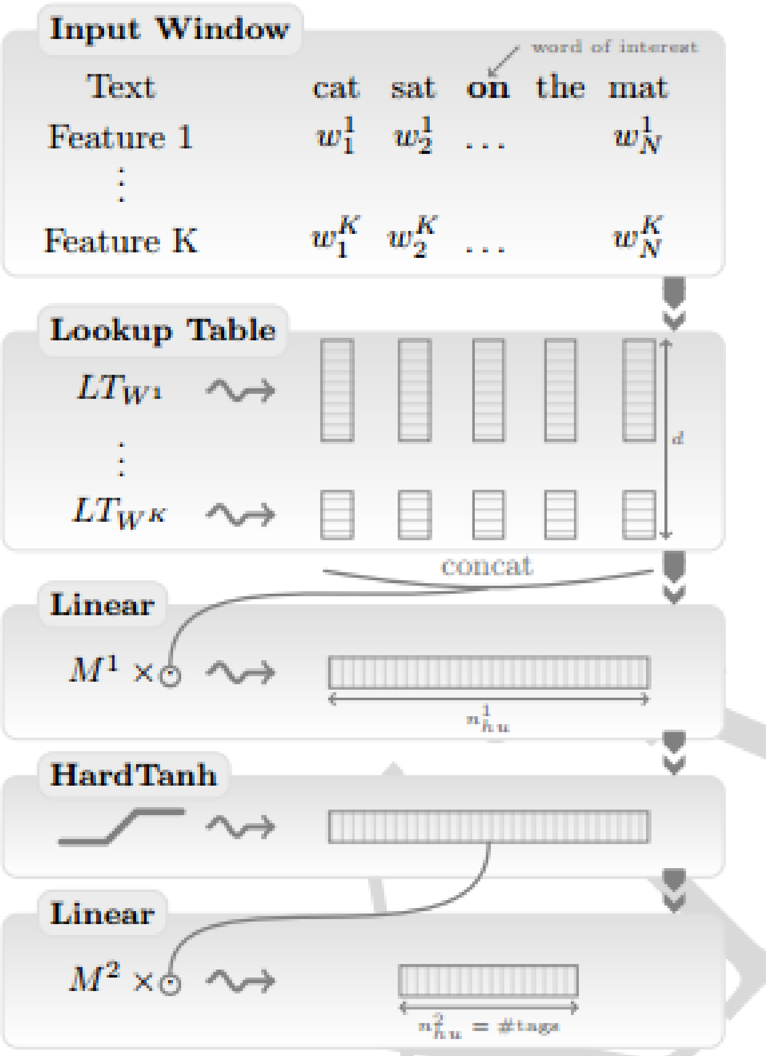
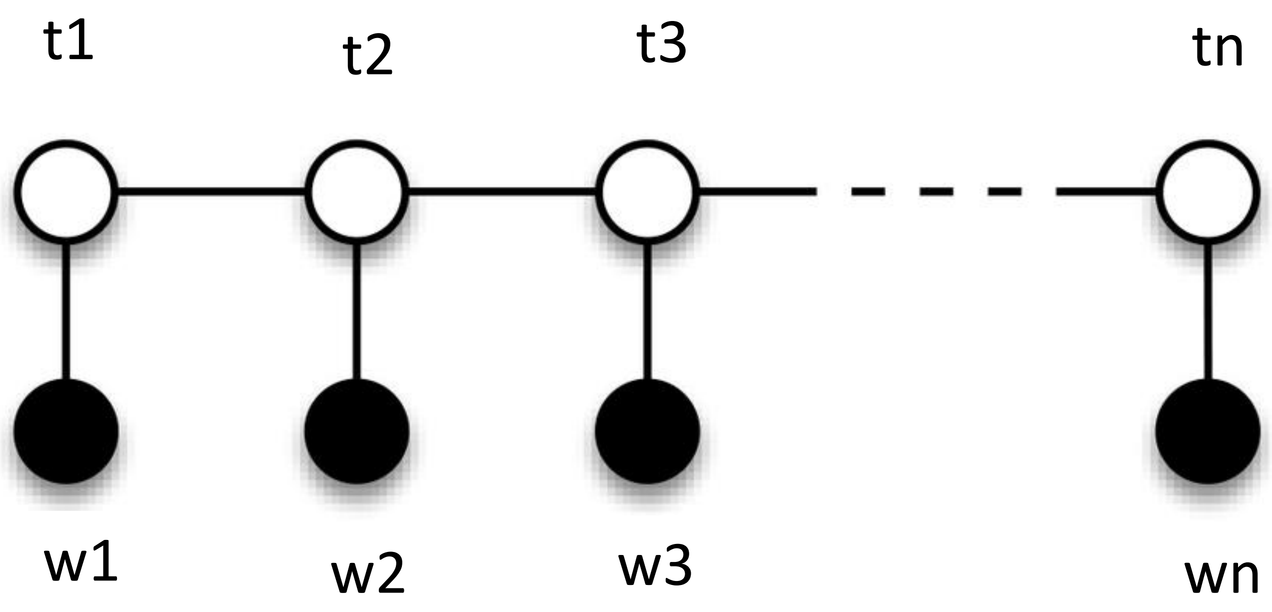


Figure 1: Window approach network.

Andor et al., 2016

- Simple feed-forward network with one hidden layer
- Conditional Random Fields (CRF) objective
- The network learns the embeddings for features
- Features:
    - are extracted from a 7 word window centered on the current word
    - The word itself, character ngrams up to length 3
    - Tags predicted for the last 4 words

Conditional Random Field - CRF

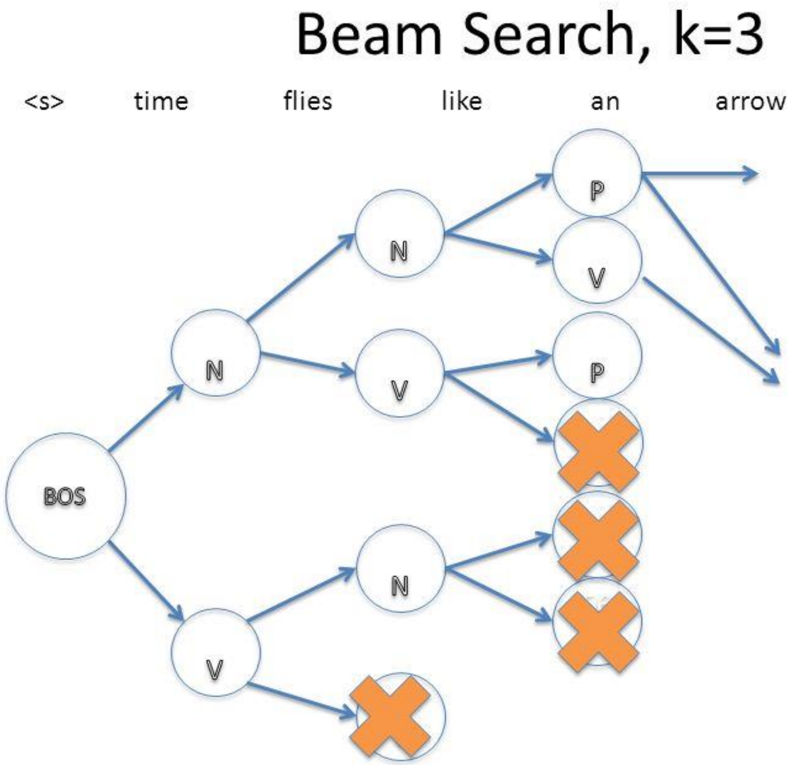


$$\hat{t}_1^n = \arg\max_{t_1^n} \frac{\exp \sum_{i=1}^{n} \rho(t_i, t_1^{i-1}, w_1^n)}{Z}$$

Figure adopted from: https://www.researchgate.net/figure/263324470_fig20_Figure-1-Linear-chain-conditional-random-fields-modelBlack-nodes-represent-observable

Beam search

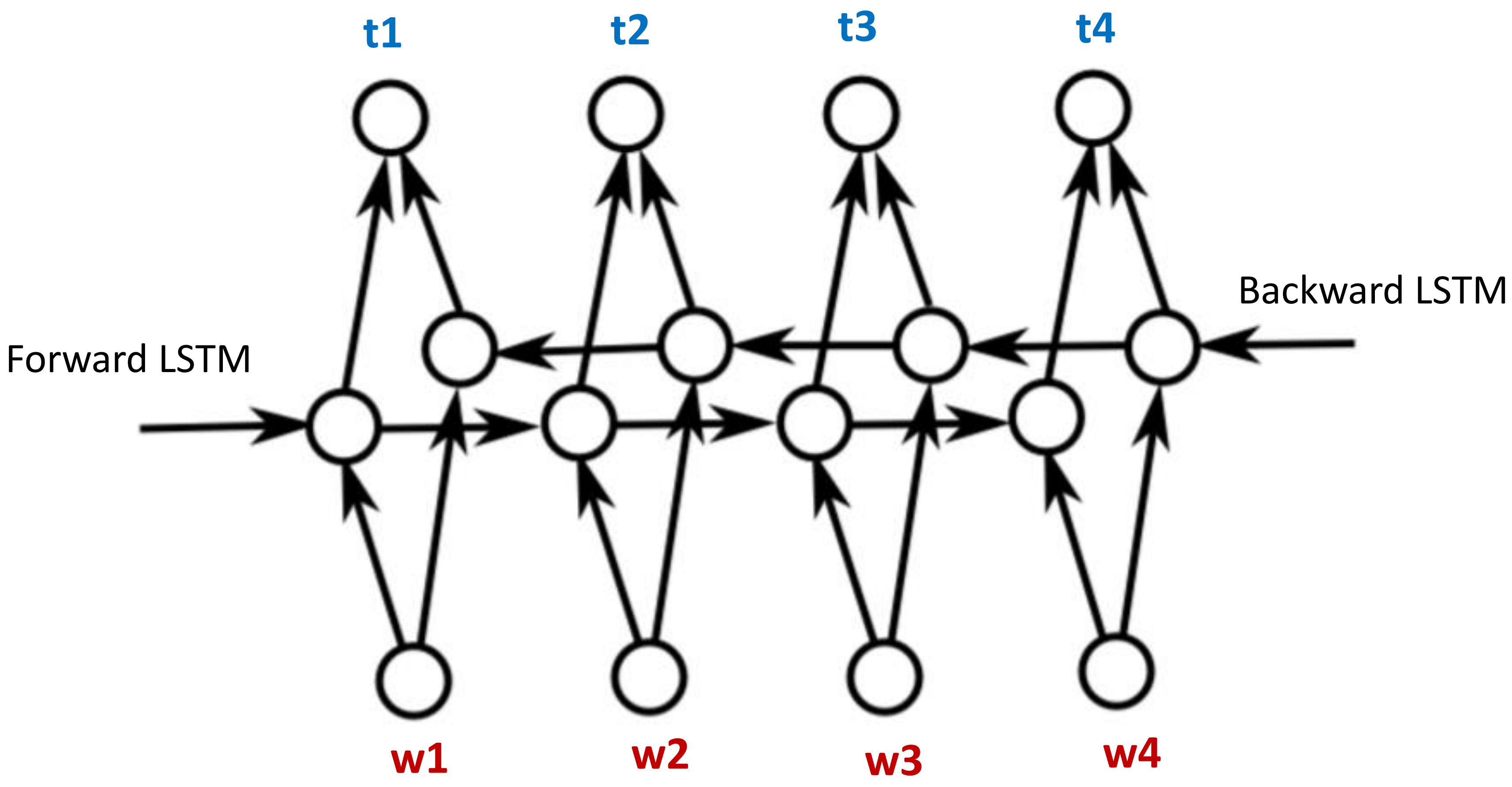
Beam search is used to approximate the normalization constant



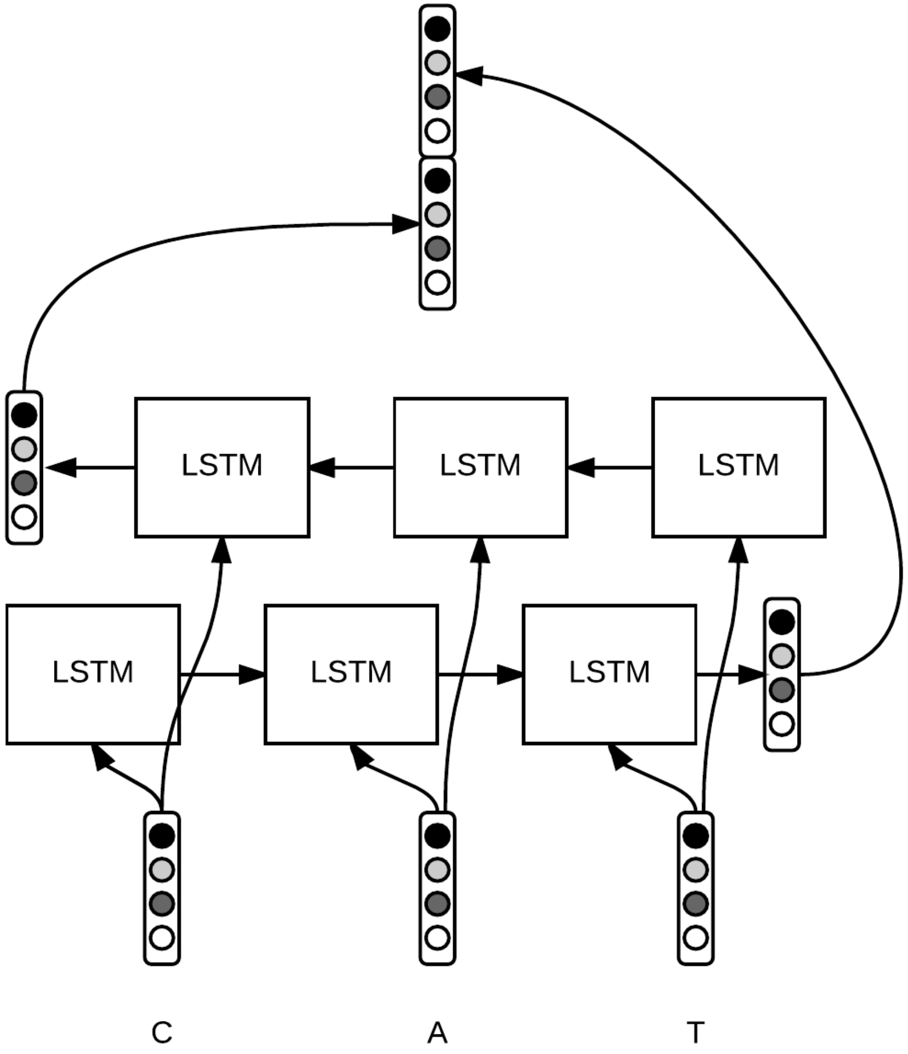
A beam search is used to efficiently select the top-N highest scoring tag sequences from among the very large set of possible tag sequences that the model can generate

http://slideplayer.com/slide/8593664/

Bidirectional LSTM



Word embeddings from characters



https://guillaumegenthial.github.io/sequence-tagging-with-tensorflow.html

Greedy softmax output layer

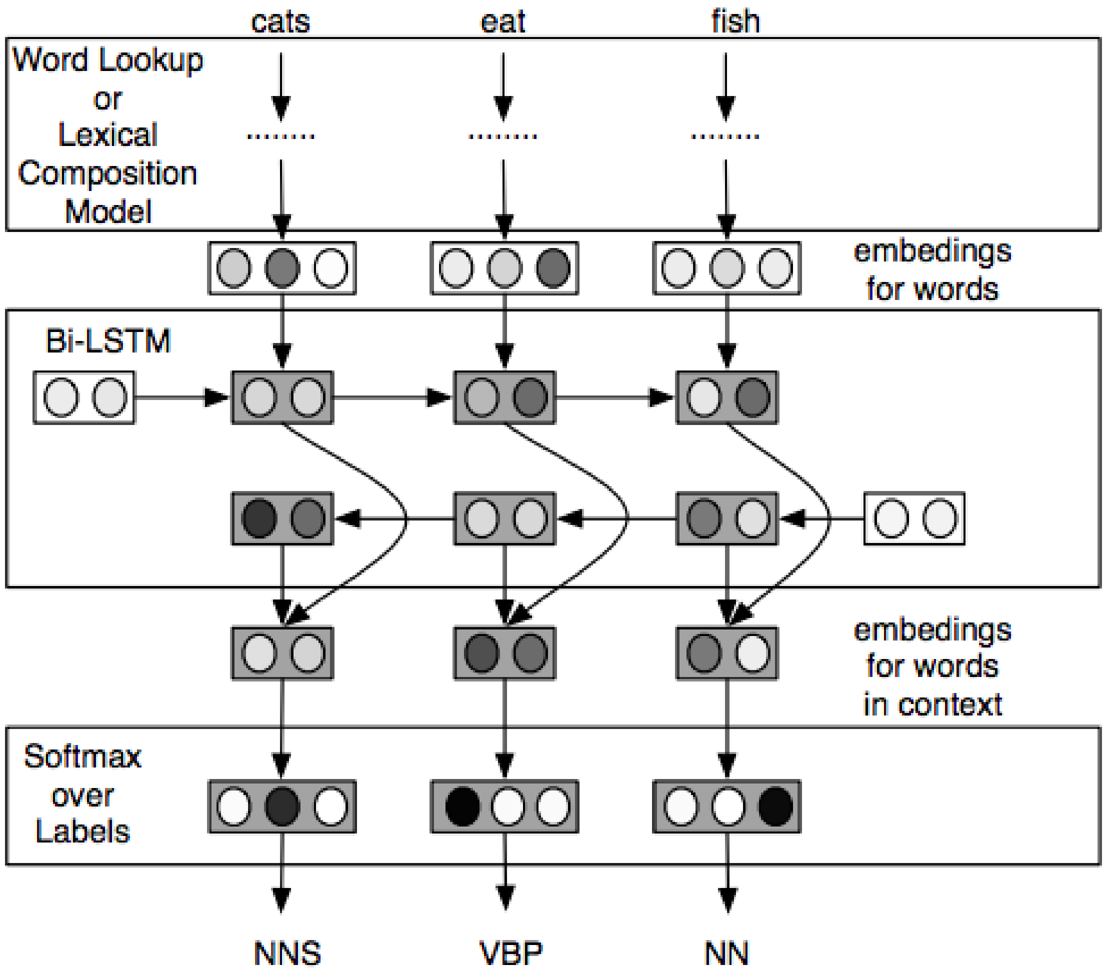
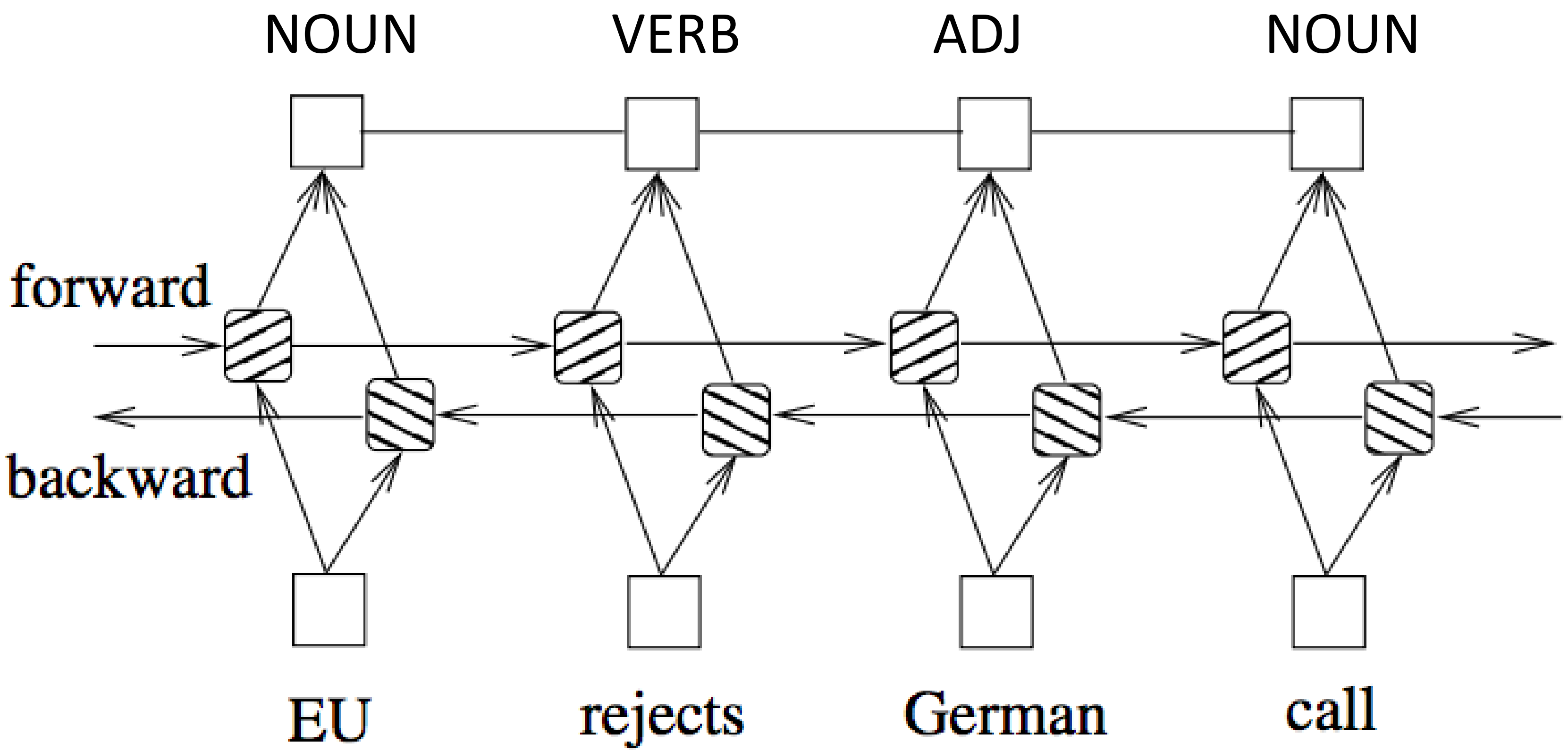


Figure 3: Ling et al., 2015. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation

CRF output layer



Adopted from Figure 7 from Huang et al., 2015. Bidirectional LSTM-CRF Models for Sequence Tagging

Conclusion

- POS tags represent syntactic functions of words
- POS tagging is the task of labelling each word in a sentence with its part-of-speech
- The classic model for POS tagging is HMM with Viterbi decoding
- The state-of-the-art models on English are log-linear models
- Lots of work recently on experimenting with different neural architectures