LINGI2263: Part-of-Speech Tagging

Pierre Dupont

Outline

1. Introduction
2. Three approaches to POS Tagging
   - Rule-Based approach
   - Probabilistic approach
   - Transformation-based tagging
3. HMM POS tagging
   - Model definition
   - HMM POS Tagging = Viterbi Decoding
   - How to estimate HMMs parameters?
4. Evaluation
5. Conclusion

Outline

1. Introduction
2. Three approaches to POS Tagging
3. HMM POS tagging
4. Evaluation
5. Conclusion

Part-Of-Speech Tagging

Sequence labeling problem

Associate a POS tag to each word token in a sentence

NNS VB TO VB AT NN IN AT NN IN JJ NN

People continue to inquire the reason for the **race** for outer space

NNP VBZ VBN TO VB NR

Secretariat is expected to **race** tomorrow

AT	article
JJ	adjective
IN	preposition
NN	singular noun
NNP	proper noun, singular
NNS	noun, plural
NR	adverbial noun
TO	to
VB	verb, base form
VBN	verb, past participle
VBZ	verb, 3sg present
...	...

45-tag Penn TreeBank tagset

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	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, singular	*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*			

Illustration from *Speech and Language Processing/2e*, D.Jurafsky and J.H.Martin, Pearson International Edition, 2009.

87-tag Brown Corpus tagset

Tag	Description	Example
(	opening parenthesis	*(, [*
)	closing parenthesis	*),]*
*	negator	*not, n't*
,	comma	*,*
–	dash	*–*
.	sentence terminator	*. ; ? !*
:	colon	*:*
ABL	pre-qualifier	*quite, rather, such*
ABN	pre-quantifier	*half, all*
ABX	pre-quantifier, double conjunction	*both*
AP	post-determiner	*many, next, several, last*
AT	article	*a, the, an, no, a, every*
BE/BED/BEDZ/BEG/BEM/BEN/BER/BEZ		*be/were/was/being/am/been/are/is*
CC	coordinating conjunction	*and, or, but, either, neither*
CD	cardinal numeral	*two, 2, 1962, million*
CS	subordinating conjunction	*that, as, after, whether, before*
DO/DOD/DOZ		*do, did, does*
DT	singular determiner	*this, that*
DTI	singular or plural determiner	*some, any*
DTS	plural determiner	*these, those, them*
DTX	determiner, double conjunction	*either, neither*
EX	existential there	*there*
HV/HVD/HVG/HVN/HVZ		*have, had, having, had, has*
IN	preposition	*of, in, for, by, to, on, at*
JJ	adjective	
JJR	comparative adjective	*better, greater, higher, larger, lower*
JJS	semantically superlative adj.	*main, top, principal, chief, key, foremost*
JJT	morphologically superlative adj.	*best, greatest, highest, largest, latest, worst*
MD	modal auxiliary	*would, will, can, could, may, must, should*
NN	(common) singular or mass noun	*time, world, work, school, family, door*
NN$	possessive singular common noun	*father's, year's, city's, earth's*
NNS	plural common noun	*years, people, things, children, problems*
NNS$	possessive plural noun	*children's, artist's parent's years'*
NP	singular proper noun	*Kennedy, England, Rachel, Congress*
NP$	possessive singular proper noun	*Plato's Faulkner's Viola's*
NPS	plural proper noun	*Americans, Democrats, Chinese*
NPS$	possessive plural proper noun	*Yankees', Gershwins' Earthmen's*
NR	adverbial noun	*home, west, tomorrow, Friday, North*
NR$	possessive adverbial noun	*today's, yesterday's, Sunday's, South's*
NRS	plural adverbial noun	*Sundays, Fridays*
OD	ordinal numeral	*second, 2nd, twenty-first, mid-twentieth*
PN	nominal pronoun	*one, something, nothing, anyone, none*
PN$	possessive nominal pronoun	*one's, someone's, anyone's*
PP$	possessive personal pronoun	*his, their, her, its, my, our, your*
PP$$	second possessive personal pronoun	*mine, his, ours, yours, theirs*
PPL	singular reflexive personal pronoun	*myself, herself*
PPLS	plural reflexive pronoun	*ourselves, themselves*
PPO	objective personal pronoun	*me, us, him*
PPS	3rd. sg. nominative pronoun	*he, she, it*
PPSS	other nominative pronoun	*I, we, they*
QL	qualifier	*very, too, most, quite, almost, extremely*
QLP	post-qualifier	*enough, indeed*
RB	adverb	
RBR	comparative adverb	*later, more, better, longer, further*
RBT	superlative adverb	*best, most, highest, nearest*
RN	nominal adverb	*here, then*

Figure 5.7 First part of original 87-tag Brown corpus tagset (Francis and Kučera, 1982). Four special hyphenated tags are omitted from this list.

Tag	Description	Example
RP	adverb or particle	*across, off, up*
TO	infinitive marker	*to*
UH	interjection, exclamation	*well, oh, say, please, okay, uh, goodbye*
VB	verb, base form	*make, understand, try, determine, drop*
VBD	verb, past tense	*said, went, looked, brought, reached, kept*
VBG	verb, present participle, gerund	*getting, writing, increasing*
VBN	verb, past participle	*made, given, found, called, required*
VBZ	verb, 3rd singular present	*says, follows, requires, transcends*
WDT	wh- determiner	*what, which*
WP$	possessive wh- pronoun	*whose*
WPO	objective wh- pronoun	*whom, which, that*
WPS	nominative wh- pronoun	*who, which, that*
WQL	how	
WRB	wh- adverb	*how, when*

Figure 5.8 Rest of 87-tag Brown corpus tagset (Francis and Kučera, 1982).

Illustrations from *Speech and Language Processing/2e*,

D.Jurafsky and J.H.Martin, Pearson International Edition, 2009.

Tag ambiguity on the Brown corpus

		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*)

- Despite having coarser tags, the 45-tag Treebank tagset is more ambiguous
- It is however most commonly used at least for evaluating automatic taggers

Illustration from *Speech and Language Processing/2e*, D.Jurafsky and J.H.Martin, Pearson International Edition, 2009.

Outline

1 Introduction

2 Three approaches to POS Tagging
- Rule-Based approach
- Probabilistic approach
- Transformation-based tagging

3 HMM POS tagging

4 Evaluation

5 Conclusion

2 Three approaches to POS Tagging
- Rule-Based approach
- Probabilistic approach
- Transformation-based tagging

Rule-based approach

1. Use a large dictionary to assign each word a set of possible tags

Word	POS
that	Adverb Pronoun Demonstrative Singular Determiner Complementizer (subordinating conjunction)

2. Apply a large list of disambiguation rules to restrict each set to a single tag for each word

Input: that
if *(next word is adjective, adverb, or quantifier)*
AND (it is followed by a sentence boundary)
AND (the previous word is not a verb like 'consider')
then eliminate non-ADV tags;
else eliminate ADV tag;

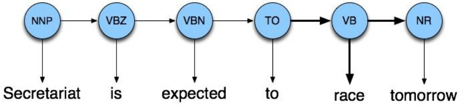
Limitations of the rule-based approach

The `EngCG` system is built on a lexicon of 56,000 entries for English word stems and includes 3,744 disambiguation rules

- Dictionary and disambiguation rules are specific to a given language
- Those linguistic resources need to be constantly updated while the language evolves

2. Three approaches to POS Tagging
   - Rule-Based approach
   - Probabilistic approach
   - Transformation-based tagging

A probabilistic approach to POS tagging



- As tagging is ambiguous, a probabilistic approach relies on the frequencies of word-tag associations in a **training corpus** to assign the tag of each word in a **new sentence**
- Choosing only the most likely tag for a given word would always assign the same tag to any word
- A better probabilistic model assigns a tag to a word according to its context: HMM POS-tagging
- When a new sentence needs to be tagged
  - the sequence of words is observed
  - the sequence of tags is hidden (= not observed)

Illustration from *Speech and Language Processing/2e*, D.Jurafsky and J.H.Martin, Pearson International Edition, 2009.

2. Three approaches to POS Tagging
   - Rule-Based approach
   - Probabilistic approach
   - Transformation-based tagging

Brill tagger

- Like rule-based approaches, each word is tagged according to some explicit rules
- Like probabilistic approaches, a disambiguation model is learned from a corpus of tagged sentences (supervised learning)

1. The most likely tag is first assigned to each word

$$P(NN \mid race) = .98 \quad P(VB \mid race) = .02$$

2. Change the current tag by applying an ordered list of transformation rules:

Change NN to VB when the previous tag is TO

Learning Transformation Rules

Given a POS tagged corpus **and** transformation templates such as:
Change tag **a** to **b** when

- the preceding (following) word is tagged **z**
- the word two before (after) is tagged **z**
- the preceding (following) word is tagged **z** and the word two before (after) is tagged **w**

Input: A tagged corpus
Output: An ordered list of transformation rules
Tag each word with its most likely tag
repeat
    Try every possible transformation by instantiating some template
    Select the one that results in the most improved tagging
    Relabel the corpus accordingly
until *stopping criterion is met*;

Pros and cons of Brill tagger

Pros

- Transformation rules can be interpreted 'linguistically'
- Learning those rules makes it possible to adapt the tagger to several languages (or language evolutions)

Limitations

- A transformation rule can be learned only if it is an instance of an abstract transformation template
- Learning is supervised only $\Rightarrow$ a tagged corpus is mandatory
- Computational complexity of learning (and, to some extent, tagging) is an issue

Outline

1 Introduction

2 Three approaches to POS Tagging

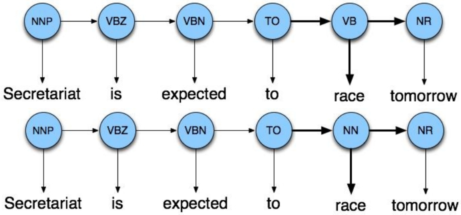
3 HMM POS tagging
- Model definition
- HMM POS Tagging = Viterbi Decoding
- How to estimate HMMs parameters?

4 Evaluation

5 Conclusion

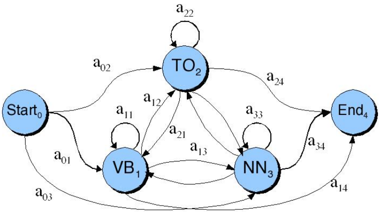
3. HMM POS tagging
   - Model definition
   - HMM POS Tagging = Viterbi Decoding
   - How to estimate HMMs parameters?

A probabilistic finite state model



- Each state is associated to a specific POS tag
- Each state emits words according to a specific emission probability distribution
- When a new sentence needs to be tagged
  - the sequence of states is hidden
  - the sequence of emitted words is observed
- Tagging a sentence reduces to find the most likely state sequence **given** the observed word sequence
  - a global criterion to tag words

First-order Markov Chain Structure



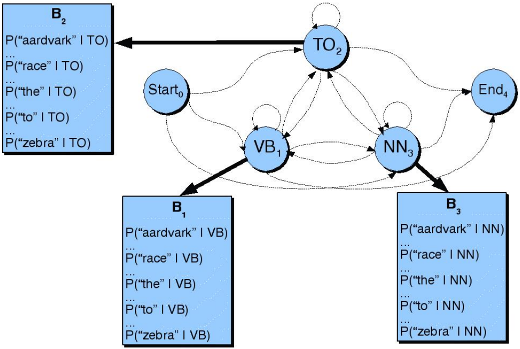
- This is equivalent to a TAG Bigram model (*Start* = <s> and *End* = </s>)
- Such a model can be estimated from a tagged corpus by counting the successive POS tags

$$a_{21} = \hat{P}(VB|TO) = \frac{C(TO, VB)}{C(TO)} \quad \text{+ some appropriate smoothing}$$

However TAGs are not observed when a new sentence need to be tagged

Illustration from *Speech and Language Processing/2e*, D.Jurafsky and J.H.Martin, Pearson International Edition, 2009.

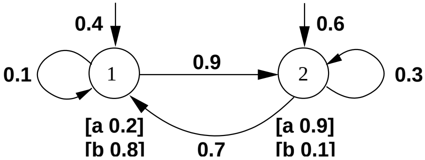
HMM Tagger



- first-order Markov Chain (= a BiGram): transition probabilities
- emission probabilities

Illustration from *Speech and Language Processing/2e*, D.Jurafsky and J.H.Martin, Pearson International Edition, 2009.

Hidden Markov Models



Definition

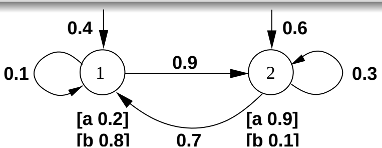
A discrete HMM (with state emission)

- W is a finite vocabulary (**a, b,...** represent the words)
- Q is a set of states
  each state **1, 2,...** is associated to a specific **POStag**
- $\mathbf{A}$ a $|Q| \times |Q|$ transition probability matrix $(\sum_{q' \in Q} \mathbf{A}_{qq'} = 1)$
- $\mathbf{B}$ a $|Q| \times |W|$ emission probability matrix $(\sum_{a \in W} \mathbf{B}_{qa} = 1)$
- π an initial probability distribution $(\sum_{q \in Q} \pi_q = 1)$
  π_q is equivalent to the transition probability a_{0q} with state 0 = `Start`

Path likelihood

The likelihood $P(s, \nu | M)$ of a sequence $s = s_1 \ldots s_{|s|}$ along a path or state sequence $\nu = q_1 \ldots q_{|s|}$ in a HMM M

$$P(s, \nu | M) = \prod_{i=1}^{|s|} P(s_i, q_i | M) = \pi_{q_1} \mathbf{B}_{q_1 s_1} \prod_{i=2}^{|s|} \mathbf{A}_{q_{i-1} q_i} \mathbf{B}_{q_i s_i}$$



$$P(abb, 122 | M) = 0.4 \times \underbrace{0.2}_{a} \times 0.9 \times \underbrace{0.1}_{b} \times 0.3 \times \underbrace{0.1}_{b}$$

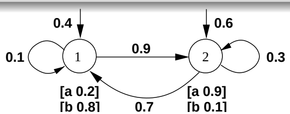
Interpretation

Probability to observe the sentence **a b b** generated by the sequence of POStags **1 2 2**

Sequence likelihood

The likelihood $P(s|M)$ of a sequence $s = s_1 \ldots s_{|s|}$ in a HMM M

$$P(s|M) = \sum_{\nu \in Q^{|s|}} P(s, \nu|M)$$



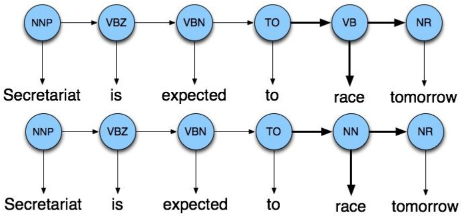
$$\begin{aligned} P(abb|M) \quad = \quad & P(abb, 111|M) + P(abb, 112|M) + P(abb, 121|M) + \\ & P(abb, 122|M) + P(abb, 211|M) + \ldots \end{aligned}$$

Interpretation

Probability to observe the sentence **a b b** generated by any sequence of 3 POStags

3. HMM POS tagging
   - Model definition
   - HMM POS Tagging = Viterbi Decoding
   - How to estimate HMMs parameters?

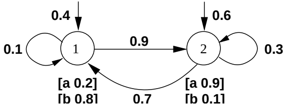
HMM POS tagging = the inverse problem



The decoding problem

Given a sentence (= sequence of words assumed to have been produced by a HMM) which is the **most likely state sequence** (= sequence of POStags) that produced it

The naïve approach



$$P(abb|M) = P(abb,111|M) + P(abb,112|M) + P(abb,121|M) + P(abb,122|M) + P(abb,211|M) + \ldots$$

Decoding

$$\nu^* = \underset{\nu}{\operatorname{argmax}}\, P(abb,\nu|M)$$

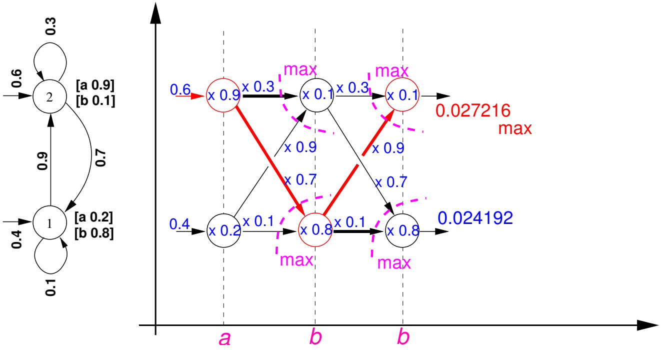
Exponential complexity

$\mathcal{O}(|Q|^{|s|})$ possible state sequences **!!**
$\mathcal{O}(45^{20}) \approx 10^{33}$ possibilities for 45 POS tags and a sentence made of 20 words

Viterbi decoding

$$\nu^* = \text{argmax}_\nu \, P(s, \nu | M)$$

Most likely state sequence for *abb* $=$ 212

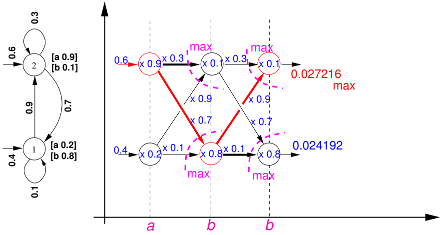


Viterbi recurrence

$$\nu^* = \mathrm{argmax}_{\nu}\, P(s, \nu | M)$$

Auxiliary quantity: $\gamma(k,t) = P(s_1 \ldots s_t, \nu_t^* = k | M)$
The probability of the most likely path ν^* reaching state k at step t

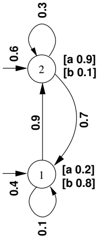
Initialization: $\gamma(k,1) = \pi_k \mathbf{B}_{ks_1}$
Recurrence: $\gamma(k,t) = \max_l [\gamma(l, t-1)\mathbf{A}_{lk}]\mathbf{B}_{ks_t}$
$back(k,t) = \mathrm{argmax}_l [\gamma(l, t-1)\mathbf{A}_{lk}]$
Termination: $P(s, \nu^* | M) = \max_l \gamma(l, |s|)$ $\quad q^*_{|s|} = \mathrm{argmax}_l\, \gamma(l, |s|)$

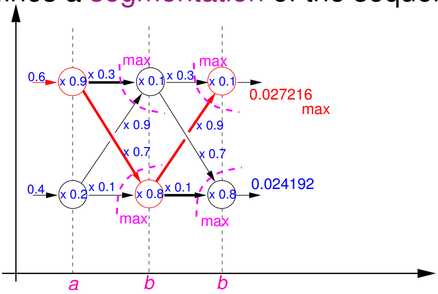


Time complexity: $\Theta(m|s|)$

Viterbi alignment

- $P(s, \nu^*)$ gives the probability of an optimal path ν^*
- Computations are done usually with log's:
  $$-\log \gamma(k,t) = \min_l[-\log \gamma(l, t-1) - \log \mathbf{A}_{lk}] - \log \mathbf{B}_{ks_t}$$
- The actual path ν^* can be recovered from the backpointers
- Time complexity is $\Theta(m|s|)$ with m the number of HMM transitions
- The path ν^* defines an alignment between states and words
- This alignment defines a segmentation of the sequence : $\underline{a}_{2} \mid \underline{b}_{1} \mid \underline{b}_{2}$





HMM POS Tagging = *Viterbi decoding*

Underlying assumptions

Given a sequence of n words $w_1^n = w_1, \ldots, w_n$ **find** a sequence of n tags $\hat{t}_1^n$ that maximises the posterior probability (MAP decision rule)

$$\begin{aligned}
\hat{t}_1^n &= \operatorname*{argmax}_{t_1^n} P(t_1^n \mid w_1^n) \\
&= \operatorname*{argmax}_{t_1^n} \underbrace{P(w_1^n \mid t_1^n)}_{likelihood} \underbrace{P(t_1^n)}_{prior} \\
&\approx \operatorname*{argmax}_{t_1^n} \underbrace{\prod_{i=1}^{n} P(w_i|t_i)}_{Hyp.1} \underbrace{\prod_{i=1}^{n} P(t_i|t_{i-1})}_{Hyp.2} = \operatorname*{argmax}_{t_1^n} \prod_{i=1}^{n} \underbrace{P(w_i|t_i)}_{Emission} \underbrace{P(t_i|t_{i-1})}_{Transition}
\end{aligned}$$

Hyp. 1: each word w_i given its tag t_i is independent of the other words and tags

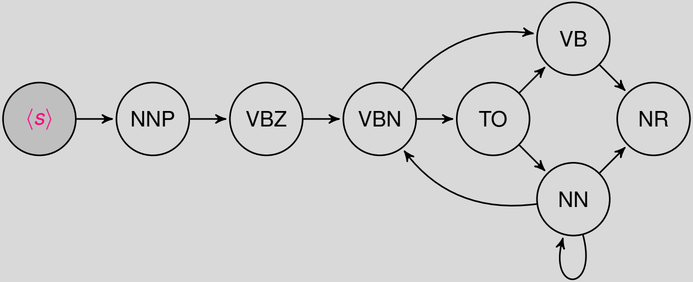
Hyp. 2: bigram tag model

3 HMM POS tagging

- Model definition
- HMM POS Tagging = Viterbi Decoding
- How to estimate HMMs parameters?

The learning problem

Given an HMM structure (by default, a fully connected graph) and several sentences (= a training corpus) to model



estimate the HMM parameters: $\mathbf{A}, \mathbf{B}(, \pi)$

Supervised learning

The learning sentences are annotated with their respective states

NNP	VBZ	VBN	TO	VB	NR
Secretariat	is	expected	to	**race**	tomorrow

HMM supervised estimation

- Build the probability estimates

$$\mathbf{B}_{ki} = \hat{P}(w_k|t_i) = \frac{C(t_i, w_k)}{C(t_i)} \quad \mathbf{A}_{kl} = \hat{P}(t_l|t_k) = \frac{C(t_k, t_l)}{C(t_k)}$$

$C(t_i, w_k)$ = number of times word w_k is observed on the POS state t_i
$C(t_k, t_l)$ = number of times POS t_k is followed by POS t_l

- Smooth the probability estimates

$$\hat{P}(w_k|t_i) = \frac{C(t_i, w_k) + \varepsilon}{C(t_i) + \sum_{w \in W} \varepsilon} \quad \hat{P}(t_l|t_k) = \frac{C(t_k, t_l) + \varepsilon'}{C(t_k) + \sum_{q \in Q} \varepsilon'}$$

$$10^{-6} \leq \varepsilon, \varepsilon' \leq 1$$

Tagging an unknown word

$$\hat{P}(w_k|t_i) = \frac{C(t_i, w_k) + \varepsilon}{C(t_i) + \sum_{w \in W} \varepsilon} \quad 10^{-6} \leq \varepsilon \leq 1$$

- **Problem:** even with a smoothed emission probability, any out-of-vocabulary word in the test set is assigned a zero probability
  - there is no Viterbi path because there is no path producing the observed sequence to be tagged
- **Usual solution:**
  - Replace any word occurring only once (or very few times) in the training set with a special marker `<UNK>`
  - Reduce the observed vocabulary accordingly and add `<UNK>` to it
  - Smooth the emission probability according to this new vocabulary

Outline

1. Introduction
2. Three approaches to POS Tagging
3. HMM POS tagging
4. Evaluation
5. Conclusion

Tagging Performance

	IN	JJ	NN	NNP	RB	VBD	VBN
IN	—	.2			.7		
JJ	.2	—	**3.3**	2.1	1.7	.2	**2.7**
NN		**8.7**	—				.2
NNP	.2	**3.3**	**4.1**	—	.2		
RB	**2.2**	2.0	.5		—		
VBD		.3	.5			—	**4.4**
VBN		**2.8**				**2.6**	—

TAG Confusion Matrix

- Each row corresponds to an actual tag
- Each column corresponds to a predicted tag
- Each entry defines the error percentage with respect to an actual tag frequency $f(i)$ (in **bold** the most common errors)
- Performance metrics:
  - Average error rate per TAG = $\frac{1}{number\ of\ rows} \sum_i (total\ of\ row_i)$
  - Tagging error rate = $\frac{\sum_i f(i)(total\ of\ row_i)}{\sum_i f(i)}$
  - Tagging accuracy = $100\% -$ Tagging error rate

Outline

1. Introduction
2. Three approaches to POS Tagging
3. HMM POS tagging
4. Evaluation
5. Conclusion

Summary

- Rule-based approaches are not adaptive to various languages or their evolutions
- Transformation-based tagging includes some learning component but
  - A tagged corpus must be provided
  - Rules are restricted from a predefined set of abstract rules
  - Computation time during training and tagging is an issue
  - No easy way to propose the N-best tagging alternatives
- HMMs offer a powerful statistical method for POS tagging
  - They can be built automatically but usually from a tagged corpus
  - N-best alternatives can be computed (not detailed here)
  - Smoothing needs to be done with some care

Your Project 2

inginious.info.ucl.ac.be/course/LINGI2263

Further reading

- Jurafsky D. and Martin J.H. (2009).
  Speech and Language Processing, 2nd edition, chapter 5, 6.
  Pearson International Edition.

- Brants, T. (2000).
  TnT – A Statistical Part-of-Speech Tagger
  Proceedings of 6th Applied Natural Language Processing Conference, p. 224–231, Seattle, Washington.