EDAN20
Language Technology
http://cs.lth.se/edan20/
Chapter 8: Part-of-Speech Tagging Using Stochastic Techniques

Pierre Nugues

Lund University
Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

September 20, 2018

Training Set

Part-of-speech taggers use a training set where every word is hand-annotated (Penn Treebank and CoNLL 2008).

Index	Word	Hand annotation	Index	Word	Hand annotation
1	Battle	JJ	19	of	IN
2	-	HYPH	20	their	PRP$
3	tested	JJ	21	countrymen	NNS
4	Japanese	JJ	22	to	TO
5	industrial	JJ	23	visit	VB
6	managers	NNS	24	Mexico	NNP
7	here	RB	25	,	,
8	always	RB	26	a	DT
9	buck	VBP	27	boatload	NN
10	up	RP	28	of	IN
11	nervous	JJ	29	samurai	FW
12	newcomers	NNS	30	warriors	NNS
13	with	IN	31	blown	VBN
14	the	DT	32	ashore	RB
15	tale	NN	33	375	CD
16	of	IN	34	years	NNS
17	the	DT	35	ago	RB
18	first	JJ	36	.	.

Part-of-Speech Tagging with Linear Classifiers

Linear classifiers are efficient devices to carry out part-of-speech tagging:

1. The lexical values are the input data to the tagger.
2. The parts of speech are assigned from left to right by the tagger.

Given the feature vector: $(w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}, t_{i-2}, t_{i-1})$, the classifier will predict the part-of-speech tag t_i at index i.

ID	FORM	PPOS	
	BOS	BOS	*Padding*
	BOS	BOS	
1	Battle	NN	
2	-	HYPH	
3	tested	NN	
...	...	...	
17	the	DT	
18	first	JJ	
19	of	IN	
20	their	PRP$	
21	countrymen	NNS	*Input features*
22	to	TO	
23	visit	VB	*Predicted tag*
24	Mexico		
25	,		
26	a		
27	boatload		
...	...	...	
34	years		
35	ago		
36	.		
	EOS		*Padding*
	EOS		

Feature Vectors

ID	Feature vectors							PPOS
	w_{i-2}	w_{i-1}	w_i	w_{i+1}	w_{i+2}	t_{i-2}	t_{i-1}	
1	BOS	BOS	Battle	-	tested	BOS	BOS	NN
2	BOS	Battle	-	tested	Japanese	BOS	NN	HYPH
3	Battle	-	tested	Japanese	industrial	NN	HYPH	JJ
...	...	...	...	...	...	...	...	...
19	the	first	of	their	countrymen	DT	JJ	IN
20	first	of	their	countrymen	to	JJ	IN	PRP$
21	of	their	countrymen	to	visit	IN	PRP$	NNS
22	their	countrymen	to	visit	Mexico	PRP$	NNS	TO
23	countrymen	to	visit	Mexico	,	NNS	TO	VB
24	to	visit	Mexico	,	a	TO	VB	NNP
25	visit	Mexico	,	a	boatload	VB	NNP	,
...	...	...	...	...	...	...	...	...
34	ashore	375	years	ago	.	RB	CD	NNS
35	375	years	ago	.	EOS	CD	NNS	RB
36	years	ago	.	EOS	EOS	NNS	RB	.

POS Annotation with the Noisy Channel Model

Modeling the problem:

$$t_1, t_2, t_3, ..., t_n \rightarrow \text{noisy channel} \rightarrow w_1, w_2, w_3, ..., w_n.$$

The optimal part of speech sequence is

$$\hat{T} = \underset{t_1, t_2, t_3, ..., t_n}{\arg\max} P(t_1, t_2, t_3, ..., t_n | w_1, w_2, w_3, ..., w_n),$$

The Bayes' rule on conditional probabilities:

$$P(A|B)P(B) = P(B|A)P(A).$$

$$\hat{T} = \underset{T}{\arg\max} P(T)P(W|T).$$

$P(T)$ and $P(W|T)$ are simplified and estimated on hand-annotated corpora, the "gold standard".

The First Term: *N*-Gram Approximation

$$P(T) = P(t_1, t_2, t_3, ..., t_n) \approx P(t_1)P(t_2|t_1)\prod_{i=3}^{n} P(t_i|t_{i-2}, t_{i-1}).$$

If we use a start-of-sentence delimiter <s>, the two first terms of the product, $P(t_1)P(t_2|t_1)$, are rewritten as
$P(<s>)P(t_1|<s>)P(t_2|<s>, t_1)$, where $P(<s>) = 1$.
We estimate the probabilities with the maximum likelihood, P_{MLE}:

$$P_{MLE}(t_i|t_{i-2}, t_{i-1}) = \frac{C(t_{i-2}, t_{i-1}, t_i)}{C(t_{i-2}, t_{i-1})}.$$

Sparse Data

If N_p is the number of the different parts-of-speech tags, there are $N_p \times N_p \times N_p$ values to estimate.
If data is missing, we can back off to bigrams:

$$P(T) = P(t_1, t_2, t_3, ..., t_n) \approx P(t_1) \prod_{i=2}^{n} P(t_i | t_{i-1}).$$

Or to unigrams:

$$P(T) = P(t_1, t_2, t_3, ..., t_n) \approx \prod_{i=1}^{n} P(t_i).$$

And finally, we can combine linearly these approximations:

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

with $\lambda_1 + \lambda_2 + \lambda_3 = 1$, for example, $\lambda_1 = 0.6$, $\lambda_2 = 0.3$, $\lambda_3 = 0.1$.

The Second Term

The complete word sequence knowing the part-of-speech sequence is usually approximated as:

$$P(W|T) = P(w_1, w_2, w_3, ..., w_n | t_1, t_2, t_3, ..., t_n) \approx \prod_{i=1}^{n} P(w_i|t_i).$$

Like the previous probabilities, $P(w_i|t_i)$ is estimated from hand-annotated corpora using the maximum likelihood:

$$P_{MLE}(w_i|t_i) = \frac{C(w_i, t_i)}{C(t_i)}.$$

For N_w different words, there are $N_p \times N_w$ values to obtain. But in this case, many of the estimates will be 0.

The POS Tagging Equation

$$\hat{T} = \arg\max_{T} P(T)P(W|T).$$

Using a bigram approximation, we have:

$$\hat{T} = P(t_1)\prod_{i=2}^{n} P(t_i|t_{i-1}) \times \prod_{i=1}^{n} P(w_i|t_i).$$

With:

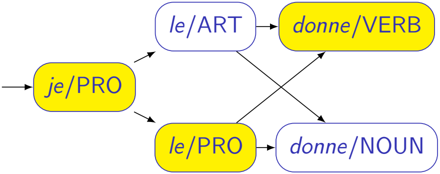
$$P_{\text{MLE}}(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

and

$$P_{\text{MLE}}(w_i|t_i) = \frac{C(w_i, t_i)}{C(t_i)}.$$

An Example

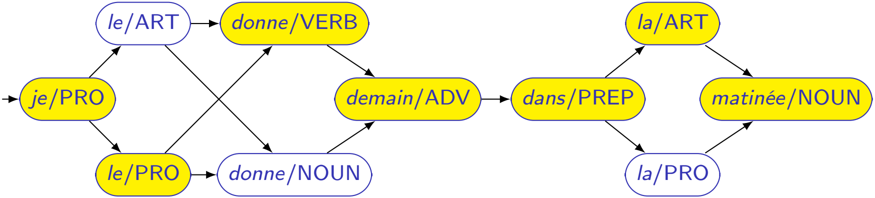
Je le donne 'I give it'



1. $P(pro|\emptyset) \times P(art|\emptyset, pro) \times P(verb|pro, art) \times P(je|pro) \times P(le|art) \times P(donne|verb)$
2. $P(pro|\emptyset) \times P(art|\emptyset, pro) \times P(noun|pro, art) \times P(je|pro) \times P(le|art) \times P(donne|noun)$
3. $P(pro|\emptyset) \times P(pro|\emptyset, pro) \times P(verb|pro, pro) \times P(je|pro) \times P(le|pro) \times P(donne|verb)$
4. $P(pro|\emptyset) \times P(pro|\emptyset, pro) \times P(noun|pro, pro) \times P(je|pro) \times P(le|pro) \times P(donne|noun)$

Viterbi (Informal)

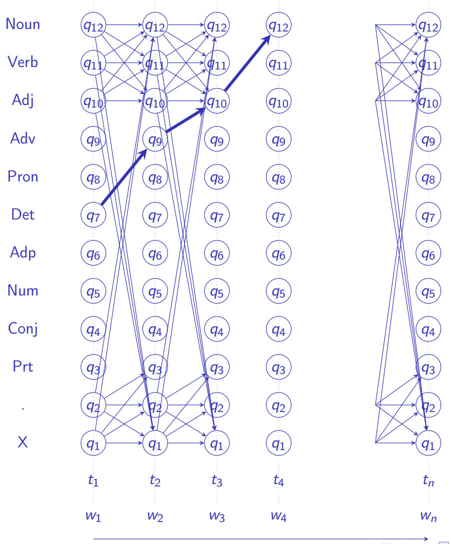
Je le donne demain dans la matinée 'I give it tomorrow in the morning'



Viterbi (Informal)

The term brought by the word *demain* has still the memory of the ambiguity of *donne*: $P(adv|verb) \times P(demain|adv)$ and $P(adv|noun) \times P(demain|adv)$.
This is no longer the case with *dans*.
According to the noisy channel model and the bigram assumption, the term brought by the word *dans* is $P(dans|prep) \times P(prep|adv)$.
It does not show the ambiguity of *le* and *donne*.
The subsequent terms will ignore it as well.
We can discard the corresponding paths.
The optimal path does not contain nonoptimal subpaths.

Trellis Representation



Filling the Trellis

$i \backslash \delta$	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8
PREP	0							
ADV	0							
PRO	0							
VERB	0							
NOUN	0							
ART	0							
<s>	1.0	0	0	0	0	0	0	0
	<s>	*Je*	*le*	*donne*	*demain*	*dans*	*la*	*matinée*

To fill the δ_3 column, for each cell j, we compute

$$\max_i P(j|i) \times P(le|j) \times \delta_2(i).$$

The pronoun cell, for instance, is filled with

$$\max_i P(\mathrm{PRO}|i) \times P(le|\mathrm{PRO}) \times \delta_2(i).$$

Worked Example in English

That round table might collapse

Looking up the words in a dictionary shows a lot of ambiguity:
What is the part of speech of *That*? determiner? pronoun? relative pronoun?
Correct tags:

*That/**determiner** round/**adjective** table/**noun** might/**modal verb** collapse/**verb**.*

Using the Penn Treebank tagset:

*That/**DT** round/**JJ** table/**NN** might/**MD** collapse/**VB**.*

Statistics from the Corpus

```
$ cut -f2,5 <corpus.txt | sort | uniq -c | grep " That\t"
 438 That DT
   5 That IN
   3 That WDT
$ cut -f2,5 <corpus.txt | sort | uniq -c | grep " round\t"
   5 round JJ
  23 round NN
   3 round VB
   1 round VBP
$ cut -f2,5 <corpus.txt | sort | uniq -c | grep " table\t"
  35 table NN
$ cut -f2,5 <corpus.txt | sort | uniq -c | grep " might\t"
 328 might MD
   4 might NN
$ cut -f2,5 <corpus.txt | sort | uniq -c | grep " collapse\t"
  57 collapse NN
   1 collapse NNP
   5 collapse VB
```

Baseline Tagger

1. Tag using the most frequent part of speech:

Words:	That	round	table	might	collapse
Tagger:	DT	NN	NN	MD	NN
Reference:	DT	JJ	NN	MD	VB

2. Evaluate your tagger:
   - Accuracy:

$$\frac{\#\text{Correct tags}}{\#\text{Tags}} = \frac{3}{5} = .6$$

   - Confusion matrix:

↓**Correct**	**Tagger** → DT	JJ	MD	NN	VB
DT	1	0	0	0	0
JJ	0	0	0	1	0
MD	0	0	1	0	0
NN	0	0	0	1	0
VB	0	0	0	1	0

Viterbi: The First Column of the Trellis

DT	0.0	§1				
IN	0.0	§2				
JJ	0.0	0.0				
MD	0.0	0.0				
NN	0.0	0.0				
NNP	0.0	0.0				
VB	0.0	0.0				
VBP	0.0	0.0				
WDT	0	§3				
<s>	1.0	0.0				
	<s>	That	round	table	might	collapse
		$P(That \mid t_1)$	$P(round \mid t_2)$	$P(table \mid t_2)$	$P(might \mid t_4)$	$P(collapse \mid t_5)$

Computing the values:

§1 $P(DT \mid BOS) \times P(That \mid DT)$

§2 $P(IN \mid BOS) \times P(That \mid IN)$

§3 $P(WDT \mid BOS) \times P(That \mid WDT)$

where $P(DT \mid BOS) = \dfrac{C(BOS, DT)}{C(BOS)}$ and $P(That \mid DT) = \dfrac{C(That, DT)}{C(DT)}$

The Rest

DT	0.0	§1	0.0	0.0	0.0	0.0
IN	0.0	§2	0.0	0.0	0.0	0.0
JJ	0.0	0.0	§4	0.0	0.0	0.0
MD	0.0	0.0	0.0	0.0	§9	0.0
NN	0.0	0.0	§5	§8	§10	§11
NNP	0.0	0.0	0.0	0.0	0.0	§12
VB	0.0	0.0	§6	0.0	0.0	§13
VBP	0.0	0.0	§7	0.0	0.0	0.0
WDT	0	§3	0.0	0.0	0.0	0.0
<s>	1.0	0.0	0.0	0.0	0.0	0.0
	<s>	That	round	table	might	collapse
		$P(That \vert t_1)$	$P(round \vert t_2)$	$P(table \vert t_2)$	$P(might \vert t_4)$	$P(collapse \vert t_5)$

The Rest: Second Column

DT	0.0	§1	0.0	0.0	0.0	0.0
IN	0.0	§2	0.0	0.0	0.0	0.0
JJ	0.0	0.0	§4	0.0	0.0	0.0
MD	0.0	0.0	0.0	0.0	§9	0.0
NN	0.0	0.0	§5	§8	§10	§11
NNP	0.0	0.0	0.0	0.0	0.0	§12
VB	0.0	0.0	§6	0.0	0.0	§13
VBP	0.0	0.0	§7	0.0	0.0	0.0
WDT	0	§3	0.0	0.0	0.0	0.0
<s>	1.0	0.0	0.0	0.0	0.0	0.0
	<s>	That	round	table	might	collapse
		$P(That\|t_1)$	$P(round\|t_2)$	$P(table\|t_2)$	$P(might\|t_4)$	$P(collapse\|t_5)$

§4 Thee competing terms:

1. $P(JJ|DT) \times §1$,
2. $P(JJ|IN) \times §2$,
3. $P(JJ|WDT) \times §3$

We take the maximum and we multiply it by $P(round|JJ)$ and we store the path.

The Rest: Second Column

DT	0.0	§1	0.0	0.0	0.0	0.0
IN	0.0	§2	0.0	0.0	0.0	0.0
JJ	0.0	0.0	§4	0.0	0.0	0.0
MD	0.0	0.0	0.0	0.0	§9	0.0
NN	0.0	0.0	§5	§8	§10	§11
NNP	0.0	0.0	0.0	0.0	0.0	§12
VB	0.0	0.0	§6	0.0	0.0	§13
VBP	0.0	0.0	§7	0.0	0.0	0.0
WDT	0	§3	0.0	0.0	0.0	0.0
<s>	1.0	0.0	0.0	0.0	0.0	0.0
	<s>	That	round	table	might	collapse
		$P(That\|t_1)$	$P(round\|t_2)$	$P(table\|t_2)$	$P(might\|t_4)$	$P(collapse\|t_5)$

§5 Three competing terms:

1. $P(NN|DT) \times §1$,
2. $P(NN|IN) \times §2$,
3. $P(NN|WDT) \times §3$

We take the maximum and we multiply it by $P(round|NN)$ and we store the path.

§6 ...

Viterbi: The Complete Algorithm

Steps	Operations
1. Initialization	$\delta_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N_p$
	$\psi_1(i) = null$
2. Induction	$\delta_{t+1}(j) = b_j(o_{t+1}) \times \max\limits_{1 \leq i \leq N_p} \delta_t(i) a_{ij}, 1 \leq j \leq N_p$, and $1 \leq t \leq n-1$
	$\psi_{t+1}(j) = \arg \max\limits_{1 \leq i \leq N_p} \delta_t(i) a_{ij}$
3. Termination	$P* = \max\limits_{1 \leq i \leq N_p} \delta_n(i)$
	$s_n* = \arg \max\limits_{1 \leq i \leq N_p} \delta_n(i)$
	The optimal path sequence is given by the backtracking:
	$s_n^*, s_{n-1}^* = \psi_n(s_n^*), s_{n-2}^* = \psi_{n-2}(s_{n-1}^*), \ldots$

Supervised Learning: A Summary

- Needs a manually annotated corpus called the **Gold Standard**
- The Gold Standard may contain errors (*errare humanum est*) that we ignore
- A classifier is trained on a part of the corpus, the **training set**, and evaluated on another part, the **test set**, where automatic annotation is compared with the "Gold Standard"
- **N-fold cross validation** is used avoid the influence of a particular division
- Some algorithms may require additional optimization on a development set
- Classifiers can use statistical or symbolic methods