

Language Technology

<http://cs.lth.se/edan20/>

Chapter 8: Part-of-Speech Tagging Using Stochastic Techniques

Pierre Nugues

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

September 17, 2020



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 | . |



Conditional Random Fields

A tagger produces a sequence of tags, where a given tag obviously depends on the previous ones.

For instance, a preposition cannot follow a determiner

We can model the tag transitions probabilities using **conditional random fields** (CRF)

The simplest form is the **linear chain**.

If \mathbf{y} denotes the output, here a sequence of tags, and \mathbf{x} , a sequence of inputs, consisting for instance of the words and the characters, we try to maximize

$$P(\mathbf{y}|\mathbf{x}),$$

i.e.

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}).$$



Conditional Random Fields (II)

As we want the output sequence to depend on the input and on previously predicted output, we rewrite $P(\mathbf{y}|\mathbf{x})$ as a joint probability, $P(\mathbf{y}, \mathbf{x})$, and, more precisely, as:

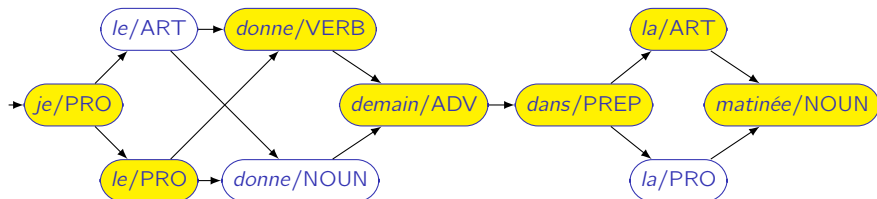
$$P(\mathbf{y}|\mathbf{x}) = \frac{P(\mathbf{y}, \mathbf{x})}{\sum_{\mathbf{y}' \in Y} P(\mathbf{y}', \mathbf{x})},$$

where \mathbf{y}' denotes a sequence, Y , the set of all the possible sequences, and $\sum_{\mathbf{y}' \in Y} P(\mathbf{y}', \mathbf{x})$ is a normalizing factor to have a sum of probabilities of one



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*: $P(\text{adv}|\text{verb}) \times P(\text{demain}|\text{adv})$ and $P(\text{adv}|\text{noun}) \times P(\text{demain}|\text{adv})$.

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 $P(\text{dans}|\text{prep}) \times P(\text{prep}|\text{adv})$.

It does not show the ambiguity of *le* and *donne*.

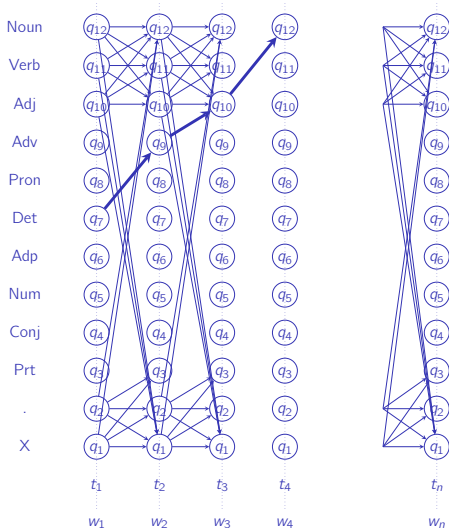
The subsequent terms will ignore it as well.

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> | <i>Je</i> | <i>le</i> | <i>donne</i> | <i>demain</i> | <i>dans</i> | <i>la</i> | <i>matinée</i> |

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

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



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

