Part of Speech Tagging

M. Rajman & J.-C. Chappelier

Laboratoire d'Intelligence Artificielle
Faculté I&C

Contents

- What is Part-of-Speech Tagging
- A simple probabilistic model: HMM tagging

Morpho-lexical level

Aims:

- resolution of some ambiguities (e.g. can:V .vs. can:N)
- suppression of some lexical variability which is not necessarily meaningful for certain applications
(e.g. difference between "*cat*" and "*cats*"in Information Retrieval).

Tools:

- Part-of-Speech tagging
- Stemming / Lemmatization

M. Rajman & J.-C. Chappelier

Lemmatization

- ☞ Automatically reduce word form to their *canonical form*, within context

canonical form: infinitive for verbs, singular for nouns, (masculin) singular for adjectives, ...

Example:

executes ⟶ execute
bought ⟶ buy

- ☞ Lemmatization is easy **if** *PoS tagging* has been performed (and lemma information is available in the lexicon)

Otherwise: "stemming" (mostly known for English: Porter's stemmer): basically, encoding most significative morphological rules

Part-of-Speech Tagging (definition)

☞ Automatically assign Part-of-Speech (PoS) Tags to words **in context**

Example:

A	computational	process	executes	programs	accurately
`Det`	`Adj`	`N`	`V`	`N`	`Adv`

Non trivial task because of lexical ambiguities:

process $\longrightarrow$ `V` or `N`?
programs $\longrightarrow$ `N` or `V`?

and of OoV forms (neologisms, proper nouns mainly).

$\Longrightarrow$ **Two** main components:

- **guesser**: assign PoS tag list to OoV
- **chooser**/disambiguator

PoS tagging (formalisation)

Given a text and a set of possible (word, tag) couples (a.k.a. the vocabulary/lexicon), choose among the possible tags for each word (known or unknown) the right one according to the context.

☞ Implies that the assertion "*the right one according to the context*" is properly defined ($\rightarrow$ goldstandard),
e.g. means "*as given by a human expert*" (!! inter-annotator agreement).

Several approaches:

- ➥ (old) Rule-based: Brill's tagger
- ➥ Probabilistic:
  Hidden Markov Models (HMM), Conditionnal Random Fields (CRF), Maximum entropy cyclic dependency network (MaxEnt)
- ➥ "Neural" (also probabilistic, but less clearly): averaged perceptrons, Support-Vector Machines (SVM), Long Short-Term Memory (LSTM)

PoS tagging (example)

Example from the Brown Corpus (`https://en.wikipedia.org/wiki/Brown_Corpus`, available in NLTK):

The/AT company/NN sells/VBZ a/AT complete/JJ line/NN of/IN gin/NN machinery/NN all/QL over/IN the/AT cotton-growing/JJ world/NN ./.

Tags explained (from original Brown Corpus documentation):

Tag	Description	Examples
AT	article	the, an, no, a, every [...]
NN	noun, singular, common	failure, burden, court, fire [...]
VBZ	verb, present tense, 3rd person singular	deserves, believes, receives, takes, [...]
JJ	adjective	recent, over-all, possible, hard-fought [...]
IN	preposition	of, in, for, by, considering [...]
QL	qualifier, pre	well, less, very, most [...]
.	sentence terminator	. ? ; ! :

Tag sets (1/2)

Complexity/Grain of tag set can vary a lot (even for the same language).

Original Brown Corpus tagset contains 87 PoS tags (!)

For instance, it contains 4 kind of adjectives:

JJ	adjective	recent, over-all, possible, hard-fought [...]
JJR	comparative adjective	greater, older, further, earlier [...]
JJS	semantically superlative adjective	top, chief, principal, northernmost [...]
JJT	morphologically superlative adjective	best, largest, coolest, calmest [...]

Tag sets (2/2)

NLTK “universal” tagset is much shorter : 12 tags (from NLTK documentation):

Tag	Meaning	Examples
ADJ	adjective	new, good, high, special, big, local
ADP	adposition	on, of, at, with, by, into, under
ADV	adverb	really, already, still, early, now
CONJ	conjunction	and, or, but, if, while, although
DET	determiner, article	the, a, some, most, every, no, which
NOUN	noun	year, home, costs, time, Africa
NUM	numeral	twenty-four, fourth, 1991, 14:24
PRT	particle	at, on, out, over per, that, up, with
PRON	pronoun	he, their, her, its, my, I, us
VERB	verb	is, say, told, given, playing, would
.	punctuation marks	. , ; !
X	other	ersatz, esprit, dunno, gr8, univeristy

Sequence Tagging

- PoS Tagging is a specific instance of a more general problem:
  the tagging of sequences

- In NLP, the considered sequences are often *word sequences*, but the nature of the targeted tagging can be very different…

Let us consider the following example:

While looking satisfied, Mary Edward Smith was disappointed.

The word sequence to tag is then:

While	looking	satisfied	Mary	Edward	Smith	was	disappointed

M. Rajman & J.-C. Chappelier

Sequence Tagging – Examples (1)

If the purpose of the tagging is to perform **Sentiment Analysis**, each of the words may be tagged by 3 possible tags:

1. tag **+** : word expressing a *positive* feeling
2. tag **-** : word expressing a *negative* feeling
3. tag **=** : word expressing a *neutral* feeling

➔

While	looking	satisfied	Mary	Edward	Smith	was	disappointed
=	=	+	=	=	=	=	-

M. Rajman & J.-C. Chappelier

Sequence Tagging – Examples (2)

If the purpose of the tagging is to perform **Named Entity Recognition** (NER), each of the words may be tagged by 2 possible tags:

1. tag **Begin_X** : first word of a Named Entity of type X
2. tag **End_X** : last word of a Named Entity of type X

➔

While	looking	satisfied	Mary	Edward	Smith	was	disappointed
			Begin _Name		**End _Name**		

M. Rajman & J.-C. Chappelier

Sequence Tagging – Examples (3)

If the purpose of the tagging is to perform **Word Sense Disambiguation** (WSD), each semantically ambiguous word may be tagged by the sense it should be associated within its specific context.

For example for ***satisfied***
1. In a state of satisfaction.
   I'm satisfied with your progress in your homework, so you can watch television now.
2. Convinced based on the available evidence.
   The judge was satisfied that the defendant did not go out with the intent to start a riot.

➔

While	looking	satisfied	Mary	Edward	Smith	was	disappointed
		satisfied_1					

Contents

- ➥ Part-of-Speech Tagging
- ☞ Probabilistic: HMM tagging

Probabilistic PoS tagging

Let $w_1^n = w_1 \dots w_n$ be a sequence of n words.

Tagging w_1^n consists in looking a corresponding sequence of Part-of-Speech (PoS) tags $T_1^n = T_1 \dots T_n$ such that the conditionnal probability $P(T_1, \dots, T_n | w_1, \dots, w_n)$ is maximal

Example:

Sentence to tag: Time flies like an arrow

Set of possible PoS tags: $\mathscr{T} = \{\texttt{Adj}, \texttt{Adv}, \texttt{Det}, \texttt{N}, \texttt{V}, \dots, \texttt{WRB}\}$

Probabilities to be compared (find the maximum):

$P(\texttt{Adj Adj Adj Adj Adj} | \text{time flies like an arrow})$

$P(\texttt{Adj Adj Adj Adj Adv} | \text{time flies like an arrow})$

$\vdots$

$P(\texttt{Adj N V Det N} | \text{time flies like an arrow})$

$\vdots$

$P(\texttt{N V Adv Det N} | \text{time flies like an arrow})$

$\vdots$

$P(\texttt{WRB WRB WRB WRB WRB} | \text{time flies like an arrow})$

(of course, many of these are null and won't even be considered)

M. Rajman & J.-C. Chappelier

Probabilistic PoS tagging

Let $w_1^n = w_1 \dots w_n$ be a sequence of n words.

Tagging w_1^n consists in looking a corresponding sequence of Part-of-Speech (PoS) tags $T_1^n = T_1 \dots T_n$ such that the conditionnal probability $P(T_1, \dots, T_n | w_1, \dots, w_n)$ is maximal

How to find $\widetilde{T_1^n} = \operatorname*{argmax}_{T_1^n} P(T_1^n | w_1^n)$?

☞ Bayes Rule:

$$P(T_1^n | w_1^n) = \frac{P(w_1^n | T_1^n) \cdot P(T_1^n)}{P(w_1^n)}$$

Probabilistic PoS tagging (2)

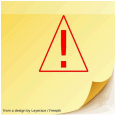
As maximization is performed for a **given** w_1^n,

$$\underset{T_1^n}{\operatorname{argmax}} P(T_1^n|w_1^n) = \underset{T_1^n}{\operatorname{argmax}} \Big(P(w_1^n|T_1^n) \cdot P(T_1^n) \Big)$$

Furthermore (chain-rule):

$$\begin{aligned} P(w_1^n|T_1^n) &= P(w_1|T_1^n) \cdot P(w_2|w_1, T_1^n) \cdot \ldots \cdot P(w_n|w_1^{n-1}, T_1^n) \\ P(T_1^n) &= P(T_1) \cdot P(T_2|T_1) \cdot \ldots \cdot P(T_n|T_1^{n-1}) \end{aligned}$$

Probabilistic PoS tagging (3)



Hypotheses:

1. limited lexical conditioning

$$P(w_i | w_1, ..., w_{i-1}, T_1, ..., T_i, ..., T_n) = P(w_i | T_i)$$

2. limited scope for syntactic dependencies: k neighbors

$$P(T_i | T_1, ..., T_{i-1}) = P(T_i | T_{i-k}, ..., T_{i-1})$$

(Note: it's a Markov assumption)

Probabilistic PoS tagging (4)

Therefore:

$$P(w_1^n|T_1^n) = P(w_1|T_1) \cdot \ldots \cdot P(w_n|T_n)$$

$$P(T_1^n) = P(T_1^k) \cdot P(T_{k+1}|T_1,\ldots,T_k) \cdot \ldots \cdot P(T_n|T_{n-k},\ldots,T_{n-1})$$

and eventually:

$$P(w_1^n|T_1^n) \cdot P(T_1^n) = P(w_1^k|T_1^k) \cdot P(T_1^k) \cdot \prod_{i=k+1}^{i=n} \left(P(w_i|T_i) \cdot P(T_i|T_{i-k}^{i-1}) \right)$$

☞ This model corresponds to a k-order Hidden Markov Model (HMM)

(order 1) Hidden Markov Models (HMM)

A order-1 HMM is:

for PoS-tagging:

- a set of states $\mathscr{C} = \{C_1, \ldots, C_m\}$ — PoS tags $\mathscr{T} = \{t^{(1)}, \ldots, t^{(m)}\}$
- a transition probabilities matrix **A**:
  $a_{ij} = P(Y_{t+1} = C_j | Y_t = C_i)$, shorten $P(C_j | C_i)$ — $P(T_{i+1} | T_i)$
- an initial probabilities vector I:
  $I_i = P(Y_1 = C_i)$ or $P(Y_t = C_i |$"start"$)$, shorten $P_I(C_i)$ — $P(T_1)$
- ☆ a set of "observables" Σ (not necessarily discreate, in general) — words $\mathscr{L} = \{a^{(1)}, \ldots, a^{(L)}\}$
- ☆ m probability densities on Σ, one for each state (*emission probabilities*):
  $B_i(o) = P(X_t = o | Y_t = C_i)$ (for $o \in \Sigma$), shorten $P(o|C_i)$ — $P(w|T_i)$

HMM will be presented in details in the next lecture

Example: PoS tagging with HMM

Sentence to tag: Time flies like an arrow

Example of HMM model:

- PoS tags: $\mathscr{T} = \{\texttt{Adj}, \texttt{Adv}, \texttt{Det}, \texttt{N}, \texttt{V}, \ldots\}$
- Transition probabilities:

  $P(\texttt{N}|\texttt{Adj}) = 0.1, P(\texttt{V}|\texttt{N}) = 0.3, P(\texttt{Adv}|\texttt{N}) = 0.01, P(\texttt{Adv}|\texttt{V}) = 0.005,$
  $P(\texttt{Det}|\texttt{Adv}) = 0.1, P(\texttt{Det}|\texttt{V}) = 0.3, P(\texttt{N}|\texttt{Det}) = 0.5$

  (plus all the others, such that stochastic constraints are fullfilled)
- Initial probabilities:

  $P_I(\texttt{Adj}) = 0.01, P_I(\texttt{Adv}) = 0.001, P_I(\texttt{Det}) = 0.1,$
  $P_I(\texttt{N}) = 0.2, P_I(\texttt{V}) = 0.003$ (+...)
- Words: $\mathscr{L} = \{an, arrow, flies, like, time, \ldots\}$
- Emission probabilities:

  $P(time|\texttt{N}) = 0.1, P(time|\texttt{Adj}) = 0.01, P(time|\texttt{V}) = 0.05, P(flies|\texttt{N}) = 0.1,$
  $P(flies|\texttt{V}) = 0.01, P(like|\texttt{Adv}) = 0.005, P(like|\texttt{V}) = 0.1, P(an|\texttt{Det}) = 0.3,$
  $P(arrow|\texttt{N}) = 0.5$ (+...)

M. Rajman & J.-C. Chappelier

Example: PoS tagging with HMM (cont.)

In this example, $12 = 3 \cdot 2 \cdot 2 \cdot 1 \cdot 1$ analyzes are possible, for example:

$P(\textit{time}/\texttt{N}\ \textit{flies}/\texttt{V}\ \textit{like}/\texttt{Adv}\ \textit{an}/\texttt{Det}\ \textit{arrow}/\texttt{N}) = 1.13 \cdot 10^{-11}$
$P(\textit{time}/\texttt{Adj}\ \textit{flies}/\texttt{N}\ \textit{like}/\texttt{V}\ \textit{an}/\texttt{Det}\ \textit{arrow}/\texttt{N}) = 6.75 \cdot 10^{-10}$

Details of one of such computation:

$$
\begin{aligned}
& P(\textit{time}/\texttt{N}\ \textit{flies}/\texttt{V}\ \textit{like}/\texttt{Adv}\ \textit{an}/\texttt{Det}\ \textit{arrow}/\texttt{N}) \\
= \quad & P_I(\texttt{N}) \cdot P(\textit{time}|\texttt{N}) \cdot P(\texttt{V}|\texttt{N}) \cdot P(\textit{flies}|\texttt{V}) \cdot P(\textit{Adv}|\texttt{V}) \cdot P(\textit{like}|\texttt{Adv}) \\
& \cdot P(\texttt{Det}|\texttt{Adv}) \cdot P(\textit{an}/\texttt{Det}) \cdot P(\texttt{N}|\texttt{Det}) \cdot P(\textit{arrow}|\texttt{N}) \\
= \quad & \text{2e-1} \cdot \text{1e-1} \cdot \text{3e-1} \cdot \text{1e-2} \cdot \text{5e-3} \cdot \text{5e-3} \cdot \text{1e-1} \cdot \text{3e-1} \cdot \text{5e-1} \cdot \text{5e-1} \\
= \quad & 1.13 \cdot 10^{-11}
\end{aligned}
$$

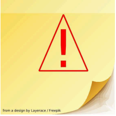
The aim is to choose the most probable tagging among the possible ones (e.g. as provided by the lexicon)

HMMs

HMM advantage: well formalized framework, efficient algorithms

- Viterbi: linear algorithm ($\mathcal{O}(n)$) that computes the sequence T_1^n maximizing $P(T_1^n|w_1^n)$ (provided the former hypotheses)
- Baum-Welch : iterative algorithm for estimating parameters from **unsupervised** data (words only, not the corresponding tag sequences) (parameters = $P(w|T_i)$, $P(T_j|T_{j-k}^{j-1})$, $P_I(T_1 \ldots T_k)$)

Parameter estimation



- supervised (i.e. manually tagged text corpus)
  Direct computation
  Problem of **missing data**
- unsupervised (i.e. raw text only, no tag)
  Baum-Welch Algorithm
  High **initial conditions sensitivity**

Good **compromise**: hybrid methods: unsupervised learning initialized with parameters from a (small) supervised learning

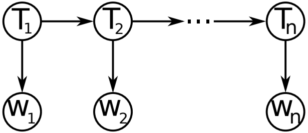
EPFL

CRF versus HMM

(linear) **Conditional Random Fields** (CRF) are a **discriminative** generalization of the HMMs where “features” no longer needs to be state-conditionnal probabilities (less constraint features).
For instance (order 1):

HMM

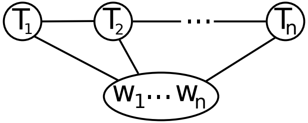
$$P(T_1^n, w_1^n) = P(T_1)\, P(w_1 | T_1) \cdot \prod_{i=2}^{n} P(w_i | T_i)\, P(T_i | T_{i-1})$$



CRF

$$P(T_1^n | w_1^n) = \prod_{i=2}^{n} P(T_{i-1}, T_i | w_1^n)$$

(with
$$P(T_{i-1}, T_i | w_1^n) \propto \exp\left(\textstyle\sum_j \lambda_j f_j(T_{i-1}, T_i, w_1^n, i)\right)$$



Other Models and Performances

[from https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art);
see also: https://nlpoverview.com/#a-pos-tagging
http://nlpprogress.com/english/part-of-speech_tagging.html]

On the “WallStreet Journal” corpus:

name	technique	publication	accuracy (%)
TnT	HMM	Brants (2000)	96.5
GENiA Tagger	MaxEnt	Tsuruoka, et al. (2005)	97.0
	Averaged Perceptron	Collins (2002)	97.1
SVMTool	SVM	Giménez and Márquez (2004)	97.2
Stanford Tagger 2.0	MaxEnt	Manning (2011)	97.3
structReg	CRF	Sun (2014)	97.4
Flair	LSTM-CRF	Akbik et al. (2018)	97.8

Keypoints

- The aim of PoS tagging is to choose among the possible tags for each word of the text the right tag according to the context
- Different **efficient** techniques exist allowing for both *supervised* and *unsupervised* learning
- Performances: 95–98 % (random $\rightarrow \simeq$ 75–90 %)
- Be familiar with the principles of HMM tagging
- Word normalization (a.k.a. “lemmatization”) is easy once PoS tagging has been done

References

[1] C. D. Manning, *Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?* In Alexander Gelbukh (ed.), Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science 6608, pp. 171–189, Springer, 2011.

[2] *Ingénierie des langues*, sous la direction de Jean-Marie Pierrel, chap. 5, Hermes, 2000.

[3] R. Dale, H. Moisl & H. Sommers, *Handbook of Natural Language Processing*, chap. 17, Dekker, 2000.

[4] C. D. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing*, chap. 10, MIT, 1999.