Computational Linguistics

Part of Speech Tagging

(the basics)

Martin Rajman

Martin.Rajman@epfl.ch

and

Jean-Cédric Chappelier

Jean-Cedric.Chappelier@epfl.ch

Laboratoire d'Intelligence Artificielle

Objectives of this lecture

➥ Present two approaches used in part-of-speech tagging

Contents

- ➥ Part-of-Speech Tagging
- ➥ Rule based: Brill algorithm
- ➥ Probabilistic: HMM tagging

Morpho-lexical level

Aims:

- ➜ resolution of some ambiguities (e.g. can:V .vs. can:N)
- ➜ reduction of the vocabulary size
- ➜ suppression of some lexical variability which is not necessarily meaningful for certain applications (e.g. in Information Retrieval).

Tools:

- ✭ Part-of-Speech tagging
- ✭ Lemmatization

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, and proper nouns mainly).

PoS tagging (formalism)

Given a text and a set of couples (word, tag) (i.e. a 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 meaningful ($\longrightarrow$ goldstandard), e.g. means "*as given by a human expert*" (!! inter-annotator agreement).

Several approaches:

- Rule-based: Brill's tagger
- Probabilistic: Hidden Markov Models (HMM), Conditionnal Random Fields (CRF), ...

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 $\longrightarrow$ execute

bought $\longrightarrow$ buy

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

Otherwise: "stemming"

Contents

- ➥ Part-of-Speech Tagging
- ☞ Rule based: Brill algorithm
- ➥ Probabilistic: HMM tagging

Brill's tagger

An "error-driven transformation-based" tagger

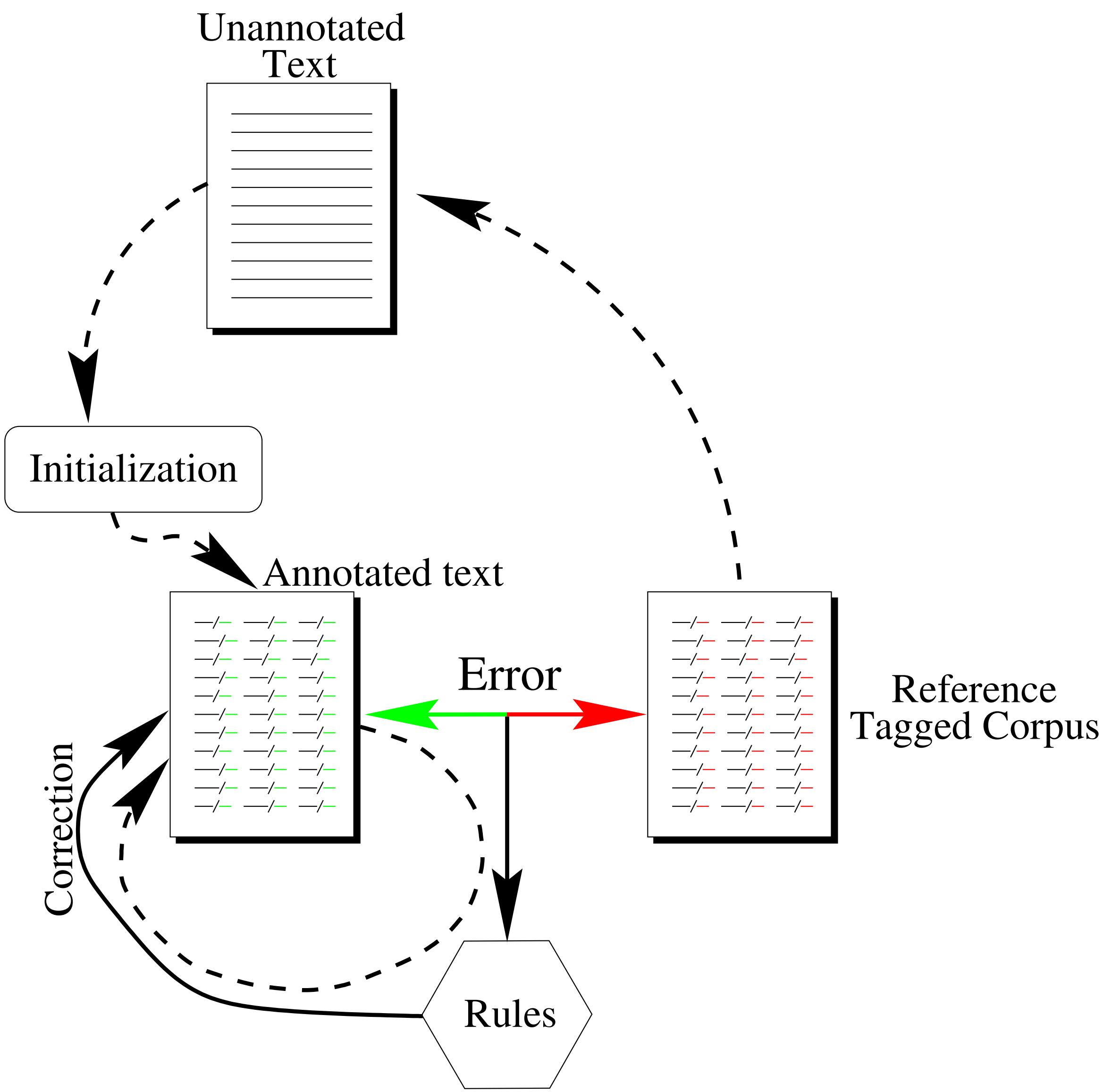
error-driven $\longrightarrow$ Supervised Learning

transformation-based $\simeq$ rules

2 distinct phases:

❶ learning phase: once, quite slow, complex

❷ application phase: many times, quick, simple


Unannotated
Text
Initialization
Annotated text
Error
Reference
Tagged Corpus
Correction
Rules

Brill's tagger: algorithm (1)

Initialization phase:

- for known words (i.e. in lexicon): the most probable tag
- for unkown words:
  - either (1992) Proper noun for Capitalized words/Noun for others
  - or (1994) Learning of guessing rules, on the same basis as contextual rules

Application phase:

- Initialize
- Apply all rules of the rule-set

Brill's tagger: algorithm (2)

Learning phase:

- Iteratively compute the error score of each candidate rule, i.e. the difference between the number of errors before and after applying the rule
- Select the best (highest score) rule, add it to the rule set and apply it to the text
- Repeat until no rule has a score above a given (predefined) threshold

☞ What are the rules to be learned? Where do they come from?

Brill's tagger (rule template)

2 types of rules:

- Rules to assign a tag to unknown words at initialization (Lexical rules)
- Rules to correct a tag in a context (Contextual rules)

Rule form:

- Lexical:

$$\text{words} \rightarrow \text{tag} \qquad \text{if Condition}$$

- Contextual:

$$\text{tag}_1 \rightarrow \text{tag}_2 \qquad \text{if Condition}$$

Brill's tagger (conditions)

Example of conditions:

- for lexical rules:
  - current word has suffix `x`, `xy`, `xyz`
  - current word has prefix `x`, `xy`, `xyz`
  - removing prefix/suffix from current word gives a known word
  - current word appears before/after word'
  - current word contains character `x`
- for contextual rules:
  - preceding/following tag (at distance one, two or three) is `Z`
  - preceding/following bigram of tags is `Y Z`
  - preceding/following word (at distance one or two) is `w`
  - current word is `w` and preceding/following word is `w'`
  - current word is `w` and preceding/following tag is `Z`

Examples of rules

Contextual Rules:

```
NN VB PREVTAG TO
VB VBP PREVTAG PRP
VBD VBN PREV1OR2TAG VBD
VBN VBD PREVTAG PRP
NN VB PREV1OR2TAG MD
VB VBP PREVTAG NNS
```

Lexical Rules:

```
NN s fhassuf 1 NNS
ed hassuf 2 VBN
ing hassuf 3 VBG
ly hassuf 2 RB
ly addsuf 2 JJ
- char JJ
```

Contents

- ➥ Part-of-Speech Tagging
- ➥ Rule based: Brill algorithm
- ☞ Probabilistic: HMM tagging

Probabilistic PoS tagging

Let $O_1^T = O_1 \ldots O_T$ be a sequence of T words.

Tagging O_1^T constists in looking a corresponding sequence of Part-of-Speech (PoS) tags $\xi_1^T = \xi_1 \ldots \xi_T$ such that the conditionnal probability $P(\xi_1, \ldots, \xi_T | O_1, \ldots, O_T)$ is maximal

How to find $\widetilde{\xi_1^T} = \underset{\xi_1^T}{\mathrm{Argmax}}\, P(\xi_1^T | O_1^T)$?

☞ Bayes Rule:

$$P(\xi_1^T | O_1^T) = \frac{P(O_1^T | \xi_1^T) \cdot P(\xi_1^T)}{P(O_1^T)}$$

Probabilistic PoS tagging (2)

As maximization is performed for a **given** O_1^T,

$$\operatorname*{Argmax}_{\xi_1^T} P(\xi_1^T | O_1^T) = \operatorname*{Argmax}_{\xi_1^T} \Big(P(O_1^T | \xi_1^T) \cdot P(\xi_1^T) \Big)$$

Furthermore (chain-rule):

$$\begin{aligned} P(O_1^T | \xi_1^T) &= P(O_1 | \xi_1^T) \cdot P(O_2 | O_1, \xi_1^T) \cdot \ldots \cdot P(O_T | O_1^{T-1}, \xi_1^T) \\ P(\xi_1^T) &= P(\xi_1) \cdot P(\xi_2 | \xi_1) \cdot \ldots \cdot P(\xi_T | \xi_1^{T-1}) \end{aligned}$$

Probabilistic PoS tagging (3)

Hypotheses:

❶ limited lexical conditioning

$$P(O_i | O_1, \ldots, O_{i-1}, \xi_1, \ldots, \xi_i, \ldots, \xi_T) = P(O_i | \xi_i)$$

❷ limited scope for syntactic dependencies: k neighbors

$$P(\xi_i | \xi_1, \ldots, \xi_{i-1}) = P(\xi_i | \xi_{i-k}, \ldots, \xi_{i-1})$$

Probabilistic PoS tagging (4)

Therefore:

$$P(O_1^T|\xi_1^T) = P(O_1|\xi_1) \cdot ... \cdot P(O_T|\xi_T)$$

$$P(\xi_1^T) = P(\xi_1^k) \cdot P(\xi_{k+1}|\xi_1, ..., \xi_k) \cdot ... \cdot P(\xi_T|\xi_{T-k}, ..., \xi_{T-1})$$

and eventually:

$$P(O_1^T|\xi_1^T) \cdot P(\xi_1^T) = P(O_1^k|\xi_1^k) \cdot P(\xi_1^k) \cdot \prod_{i=k+1}^{i=n} \left(P(O_i|\xi_i) \cdot P(\xi_i|\xi_{i-k}^{i-1}) \right)$$

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

Hidden Markov Models (HMM)

- a set of states $\mathcal{C} = \{C_1, ..., C_n\}$ PoS tags
- a transition probabilities matrix $\mathbf{A}$:

  $a_{ij} = P(\xi_{t+1} = C_j | \xi_t = C_i)$, shorten $P(C_j|C_i)$
- an initial probabilities vector I:

  $I_i = P_I(\xi_1 = C_i)$, shorten $P_I(C_i)$
- an alphabet Σ (not necessarily finite) words
- n probability densities on Σ (*emission probabilities*):

  $B_i(w) = P(O_t = w | \xi_t = C_i)$ (for $w \in \Sigma$), shorten $P(w|C_i)$.

HMM will be presented in details in the next lecture

HMMs

HMM Advantage: well formalized framework, efficient algorithms

- Viterbi: linear algorithm ($\mathcal{O}(n)$) that computes the sequence ξ_1^T maximizing $P(\xi_1^T|O_1^T)$ (provided the former hypotheses)
- Baum-Welch : iterative algorithm for estimating parameters from unsupervised data (words only, not the corresponding tag sequences)
  (parameters = $P(w|C_i)$, $P(C_j|C_{i_1}^{i_k})$, $P_I(C_{i_1}...C_{i_k})$)

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

Improvements: Discriminative methods (CRF, averaged perceptrons [Collins 2002]), semi-supervised (averaged perceptrons)

Tagging: conclusion

- ✌ efficient algorithms
- ☞ Learning abilities
- ✏ Performances: 95–98 % (random $\longrightarrow \simeq$ 75–90 %)

Keypoints

- 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 techniques used in tagging, for example, probabilistic approach, rule-based approach
- Be familiar with principles of Brill algorithm and HMM tagging

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.