

# Tagging

Regina Barzilay  
EECS Department  
MIT

November 15, 2004

---

## Last Time

- Language modeling:
  - n-gram models
  - LM evaluation
- Smoothing
  - Discounting
  - Backoff
  - Interpolation

---

## Tagging

**Task:** Label each word in a sentence with its appropriate part of speech

**Input:** *Our enemies are innovative and resourceful , and so are we. They never stop thinking about new ways to harm our country and our people, and neither do we.*

**Output:** *Our/PRP\$ enemies/NNS are/VBP innovative/JJ and/CC resourceful/JJ ,/, and/CC so/RB are/VB we/PRP ?/? . They/PRP never/RB stop/VB thinking/VBG about/IN new/JJ ways/NNS to/TO harm/VB our/PROP\$ country/NN and/CC our/PRP\$ people/NN, and/CC neither/DT do/VB we/PRP*

---

## Motivation

- Part-of-speech(POS) tagging is important for many applications
  - Parsing
  - Language modeling
  - Q&A and Information extraction
  - Text-to-speech
- Tagging techniques can be used for a variety of tasks
  - Semantic tagging
  - Dialogue tagging

## How to determine the tag set?

“The definition [of the parts of speech] are very far from having attained the degree of exactitude found in Euclidean geometry” *Jespersen, The Philosophy of Grammar*

- Agreement on coarse lexical categories (at least, for some languages)
  - Closed class: prepositions, determiners, pronouns, particles, auxiliary verbs
  - Open class: nouns, verbs, adjectives and adverbs
- Multiple tag sets of various granularity
  - Penn tag set (45 tags), Brown tag set (87 tags), CLAWS2 tag set (132 tags)

Tagging

4/31

## Is Tagging Hard?

“Time flies like an arrow”

- Many words may appear in several categories
- However, most words appear predominantly in one category
  - “Dumb” tagger which assigns the most common tag to each word achieves 90% accuracy (Charniak et al., 1993)
  - Are we happy with 90%?

Tagging

6/31

## Penn Tree Tags

Tag	Description	Example
CC	conjunction	and, but
DT	determiner	a, the
JJ	adjective	red
NN	noun, sing.	rose
RB	adverb	quickly
VBD	verb, past tense	grew

Tagging

5/31

## Information Sources in Tagging

- Lexical: look at word itself

Word	Noun	Verb	Preposition
flies	21	23	0
like	10	30	21

- Syntagmatic: look at nearby words
  - What is more likely: “DT JJ NN” or “DT JJ VBP”?

Tagging

7/31

## Learning to Tag

---

- Transformation-based Learning
- Hidden Markov Model Taggers
- Log-linear models

Tagging

8/31

## Transformations

---

- Rewrite rule:  $tag^1 \rightarrow tag^2$ , if  $C$  holds.
  - Templates are hand-selected.
- Triggering environment ( $C$ ):
  - tag-triggered
  - word-triggered
  - morphology-triggered

Tagging

10/31

## Transformation-based Learning (TBL)

---

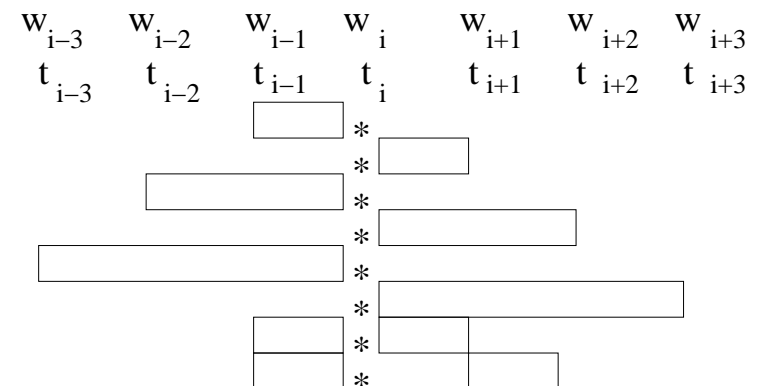
- TBL is “in between” symbolic and corpus-based methods
- TBL exploit a wider range of lexical and syntactic regularities (very few parameters to estimate)
- Key TBL components:
  - a specification of which “error-correcting” transformations are admissible
  - the learning algorithm

Tagging

9/31

## Transformation Templates

---



Tagging

11/31

## Example of Transformations

---

Source Tag	Target Tag	Triggering environment
NN	VB	previous tag is TO
VBP	VB	one of the previous tags is MD
JJR	JJR	next tag is JJ
VBP	VB	one of the prev. two words is "n't"

Tagging

12/31

## Algorithm

---

Notations:  $C_k$  — corpus tagging at iteration  $k$ ,  $E(C_k)$  — the number of mistakes in tagged corpus  $E(C_k)$

$C_0$  := corpus with each word tagged with its most frequent tag

**for**  $k := 0$  **step 1 do**

$v :=$  the transformation  $u_i$  that minimizes  $r(u_i(C_k))$

**if**  $(E(C_k) - E(v(C_k))) < \epsilon$  **then break fi**

$C_{k+1} := v(C_k)$

$\tau_{k+1} := \tau$

**end**

Output sequence:  $\tau_1, \dots, \tau_n$

Tagging

14/31

## Learning component of TBL

---

Greedy search for the optimal sequence of transformations

- Select the best transformations
- Determine their order of applications

Tagging

13/31

## Initialization

---

- Alternative approaches
  - random
  - most frequent tag
  - ...
- In practice, TBL is not sensitive to the original assignment

Tagging

15/31

## Rule Application

---

- Left-to-right order of application
- Immediate vs delayed effect:  
Consider “A → B if the preceding tag is A”
  - Immediate: AAAA → ?
  - Delayed: AAAA → ?

Tagging

16/31

## The Tagger

---

- Input
  - untagged data
  - rules (S) learned by the learner
- Tagging
  - use the same initialization as the learner did
  - apply all the learned rules (keep the proper order of application)
  - the last intermediate data is the output

Tagging

18/31

## Rule Selection

---

- We select both the template, and its instantiation.
- Each rule  $\tau$  modifies given annotations
  - improves in some places  $c_{improved}(\tau)$
  - worsens in some places  $c_{worsened}(\tau)$
  - does not touch the remaining data
- The contribution of the rule is  
 $c_{improved}(\tau) - c_{worsened}(\tau)$
- Rule selection at iteration  $i$   
 $\tau_{selected}(i) = \operatorname{argmax}_{\tau} \operatorname{contrib}(\tau)$

Tagging

17/31

## Discussion

---

- What is the time complexity of TBL?
- Is it possible to develop an unsupervised TBL tagger?

Tagging

19/31

## Relation to Other Models

---

- Probabilistic models:
  - “k-best” tagging
  - encoding of prior knowledge
- Decision Trees
  - TBL is more powerful (Brill, 1995)
  - TBL is immune to overfitting

Tagging

20/31

## Parameter Estimation

---

- Apply chain rule:

$$P(T, S) = \prod_{j=1}^n P(T_j | S_1, \dots, S_{j-1}, T_1, \dots, T_{j-1}) * P(S_j | S_1, \dots, S_{j-1}, T_1, \dots, T_j)$$

- Assume independence (Markov assumption):

$$= \prod_{j=1}^n P(T_j | T_{j-2}, T_{j-1}) * P(S_j | T_j)$$

Tagging

22/31

## Markov Model

---

Intuition: Pick the most likely tag for each word of a sequence

- We will model  $P(T, S)$ , where  $T$  is a sequence of tags, and  $S$  is a sequence of words

- $P(T|S) = \frac{P(T,S)}{\sum_T P(T,S)}$

$$\text{Tagger}(S) = \operatorname{argmax}_{T \in T^n} \log P(T|S) = \operatorname{argmax}_{T \in T^n} \log P(T, S)$$

Tagging

21/31

## Example

---

*They/PRP never/RB stop/VB thinking/VBG about/IN new/JJ ways/NNS to/TO harm/VB our/PROP\$ country/NN and/CC our/PRP\$ people/NN, and/CC neither/DT do/VB we/PRP.*

$$P(T, S) = P(\text{PRP}|S, S) * P(\text{They}|\text{PRP}) * P(\text{RB}|S, \text{PRP}) * P(\text{never}|\text{RB}) * \dots$$

Tagging

23/31

## Estimating Transition Probabilities

---

$$P(T_j|T_{j-2}, T_{j-1}) = \lambda_1 * \frac{\text{Count}(T_{j-2}, T_{j-1}, T_j)}{\text{Count}(T_{j-2}, T_{j-1})} + \lambda_2 * \frac{\text{Count}(T_{j-1}, T_j)}{\text{Count}(T_{j-1})} + \lambda_3 * \frac{\text{Count}(T_j)}{\text{Count}(\sum_i T_i)}$$

Tagging

24/31

## Dealing with Low Frequency Words

---

- Split vocabulary into two sets
  - Frequent words — words occurring more than 5 times in training
  - Low frequency words — all other words
- Map low frequency words into a small, finite set, depending on prefixes, suffixes etc. (see Bikel et al., 1998)

Tagging

26/31

## Estimating Emission Probabilities

---

$$P(S_j|T_j) = \frac{\text{Count}(S_j, T_j)}{\text{Count}(T_j)}$$

Problem: unknown or rare words

- Proper names
  - “King Abdullah of Jordan, the King of Morocco, I mean, there’s a series of places — Qatar, Oman – I mean, places that are developing — Bahrain — they’re all developing the habits of free societies.”
- New words
  - “They underestimated me.”

Tagging

25/31

## Efficient Tagging

---

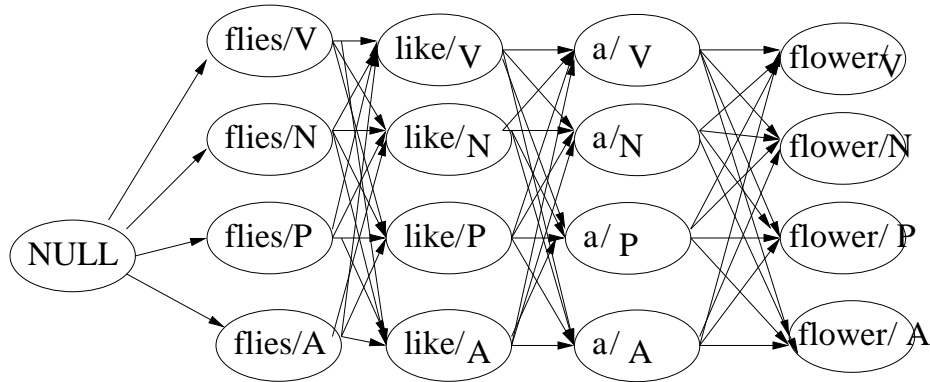
How to find the most likely a sequence of tags for a sequence of words?

- The brute force search is dreadful — for  $N$  tags and  $W$  words, the cost is  $N^W$
- Idea: use memoization (the Viterbi Algorithm)
  - Sequences that end in the same tag can be collapsed together since the next tag depends only on the current tag of the sequence

Tagging

27/31

## Efficient Tagging



Tagging

28/31

## Performance

- HMM taggers are very simple to train
- Perform relatively well (over 90% performance on named entities)
- Main difficulty is modeling of  $p(word|tag)$

Tagging

30/31

## The Viterbi Algorithm

- **Base case:**

$$\pi[0, START] = \log 1 = 0$$

$$\pi[0, t_{-1}] = \log 0 = \infty$$

for all other  $t_{-1}$

- **Recursive case:** for  $i = 1 \dots S.length$ , for all  $t_{-1} \in T$ :

$$\pi[i, t_{-1}] = \max_{t \in T \cup START} \{\pi[i-1, t] + \log P(t_{-1}|t) + \log P(S_i|t_{-1})\}$$

Backpointers allow us to recover the max probability sequence:

$$BP[i, t_{-1}] = \operatorname{argmax}_{t \in T \cup START} \{\pi[i-1, t] + \log P(t_{-1}|t) + \log P(S_i|t_{-1})\}$$

Tagging

29/31

## Conclusions

- Tagging is relatively easy task (at least, in a supervised framework, and for English)
- Factors that impact tagger performance include:
  - The amount of training data available
  - The tag set
  - The difference in vocabulary between the training and the testing
  - Unknown words
- TBL and HMM framework can be used for other tasks

Tagging

31/31