Introduction to Human Language Technologies
4. POS tagging

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Outline

Outline

Goal

- Morphological analysis provides lexical information related to forms (POS, num, gen, tense, . . .)
- Multiple analyses can result (POS tags from Penn Treebank tagset)

form	analyses	example of use
fish	NNS	'Cats eat fish'
	VBG	'I am fishing'
bass	NN	'I saw you play the bass'
	JJ	'Bass clarinets sound good'

- **Goal:** disambiguate POS of word forms occurring in text

Motivation

Examples of applications of POS tagging:

- Syntactic parsing: words with the same POS tag play a similar syntactic role

  Ex: a determiner followed by a common noun is a noun phrase

- Machine translation

  Ex: (POS tags from Penn Treebank tagset)

'El hombre	**bajo**	toca el	**bajo**	**bajo**	el puente'
POS tagging	NN JJ IN VB		NN JJ IN VB	NN JJ IN VB	
possible English words	low small short poor		bass	under below	
'The	**small**	man plays the	**bass**	**under**	the bridge'

Outline

Open class vs. Closed class

- General classes:
  - **Closed class**: never invent new closed items (functional words)
    Usual subclasses for indo-european languages:

    prepositions, conjunctions, determiners, pronouns, auxiliary verbs or particles (prepositions or adverbs in phrasal verbs)
  - **Open class**: new open items can be invented
    Usual subclasses for indo-european languages:

    nouns, non-auxiliary verbs, adjectives and adverbs
- Each language defines its particular set of subclasses
- Subclasses can be represented with a particular granularity by a set of categories

  Ex: Brown corpus: annotated with 87 different POS tags

  Ex: Penn Treebank corpus: with 45 different POS tags

Penn Treebank tagset

CC	Coordinating conjunction	PP	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition	SYM	Symbol
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund
NN	Noun, singular	VBN	Verb, past participle
NNP	Proper noun, singular	VBP	Verb, non-3rd ps. sing. present
NNS	Noun, plural	VBZ	Verb, 3rd ps. sing. present
NNPS	Proper noun, plural	WDT	wh-determiner
PDT	Predeterminer	WP	wh-pronoun
POS	Posessive ending	WP	Possessive wh-pronoun
PRP	Personal pronoun	WRB	wh-adverb

12 categories more related to punctuation marks

Ex: to/TO give/VB priority/NN to/IN teacher/NN pay/NN rises/NNS

Outline

POS tagging methods

Frequently used methods:

- Rule-based methods:
  - Rules built manually are not frequently used. High production cost
  - Rules learnt automatically from training corpus.

    Ex: Brill's tagger.
- Stochastic methods:
  - Based on Hidden Markov Models learnt automatically from training corpus.

Outline

Stochastic taggers

Goal: Assign the most likely POS-tag sequence to a word sequence.

$W = w_1 \dots w_n$ (a word sequence)
$T = t_1 \dots t_n$ (a POS-tag sequence)

Tagger result: $\hat{T} = \operatorname*{argmax}_{T} P(T|W)$

1. How is $P(T|W)$ computed?
   Apply a Hidden Markov Model
2. How is $\hat{T}$ found?
   Apply Viterbi algorithm

Outline

Preliminaries: Markov model

- $X = (X_1, \ldots, X_T)$ sequence of random variables taking values in observed states $S = \{s_1, \ldots, s_N\}$
- Inference: Sequence probability $P(X)$?
- Markov Properties
  - Limited Horizon:
    $P(X_{t+1} = s_k \mid X_1, \ldots, X_t) = P(X_{t+1} = s_k \mid X_t)$
  - Time Invariant (Stationary):
    $P(X_{t+1} = s_k \mid X_t) = P(X_2 = s_k \mid X_1)$
- Transition matrix:
  $a_{ij} = P(X_{t+1} = s_j \mid X_t = s_i); \quad \forall i, j \; a_{ij} \geq 0; \quad \forall i \; \sum_{j=1}^{N} a_{ij} = 1$
- Initial probabilities (or extra state s_0):
  $\pi_i = P(X_1 = s_i); \quad \sum_{i=1}^{N} \pi_i = 1$

Preliminaries: Markov model

POS tagging

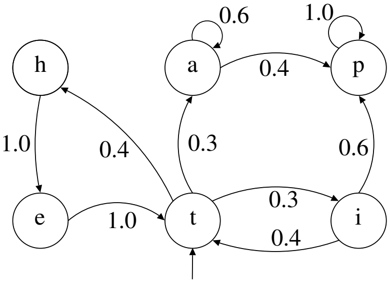
POS Taggers

Hidden Markov Model

Sequence probability: (Bayesian rule+limited horizon)

$$
\begin{aligned}
P(X_1, .., X_T) = & \\
& = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1 X_2) \ldots P(X_T \mid X_1..X_{T-1}) \\
& = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_2) \ldots P(X_T \mid X_{T-1}) \\
& = \pi_{X_1} \prod_{t=2}^{T} a_{X_{t-1}X_t}
\end{aligned}
$$

Example:



$P(t, h, e, t, i, p, p) = 1 \cdot (0.4 \cdot 1 \cdot 1 \cdot 0.3 \cdot 0.6 \cdot 1 \cdot 1) = 0.42$

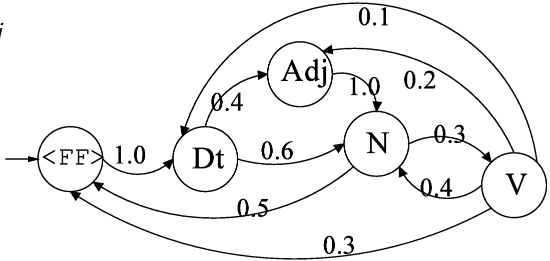
Hidden Markov model

- $X = (X_1, \ldots, X_T)$ sequence of random variables taking values in unobserved [hidden] states $S = \{s_1, \ldots, s_N\}$ given a sequence of observations $O = (O_1, \ldots, O_T)$
- Inference: Probability of ...
  - a process: $P(O)$?
  - the state of a process at the end: $P(X_T \mid O)$?
  - the explanation of a process: $P(X_1, \ldots, X_T \mid O)$?
    POS tagging: X = POS tags; O = words
- Transition matrix:
  $a_{ij} = P(X_{t+1} = s_j \mid X_t = s_i); \quad \forall i, j \; a_{ij} \geq 0; \quad \forall i \; \sum_{j=1}^{N} a_{ij} = 1$
- Initial probabilities (or extra state s_0):
  $\pi_i = P(X_1 = s_i); \quad \sum_{i=1}^{N} \pi_i = 1$
- Emission Probability:
  $b_{ik} = P(O_t = k \mid X_t = s_i) \quad \forall i, k \; b_{ik} \geq 0; \quad \forall i \; \sum_{k=1}^{N} b_{ik} = 1$

Hidden Markov model

Example (horizon=1; bigrams)

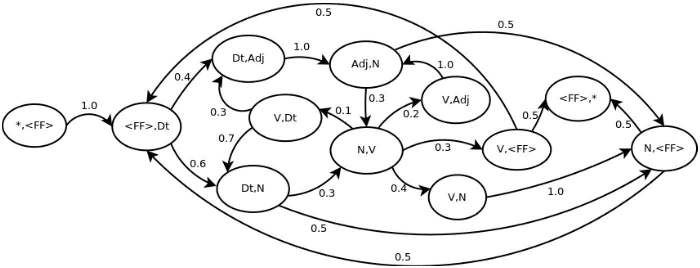
a_{ij}



b_{ik}	.	the	this	cat	kid	eats	runs	fish	fresh	little	big
<FF>	1.0										
Dt		0.6	0.4								
N				0.3	0.1		0.1	0.3	0.2		
V					0.1	0.5	0.3	0.1			
Adj					0.1				0.2	0.3	0.4

Hidden Markov model

Example (horizon=2; trigrams)



b_{ik}	.	the	this	cat	kid	eats	runs	fish	fresh	little	big
?,<FF>	1.0										
?,Dt		0.6	0.4								
?,N				0.3	0.1		0.1	0.3	0.2		
?,V					0.1	0.5	0.3	0.1			
?,Adj					0.1				0.2	0.3	0.4

Learning of parameters

- Parameters a_{ij}, b_{ik} and π_i can be estimated over a training corpus C
- Use smoothing techniques
- Use Baum-Welch algorithm
- **learning of parameters will be studied in AHLT**

Learning of parameters

Example: MLE estimator; u, v, w different POS tags in the training corpus

- bigram-based HMM

$$a(u,v) \approx P_{MLE}(v \mid u) = \frac{c(u,v)}{c(u)}$$

$$b(O_i,u) \approx P_{MLE}(O_i \mid u) = \frac{c(u,O_i)}{c(u)}$$

$$\pi(u) \approx P_{MLE}(u \mid *) = \frac{c(*,u)}{c(*)}$$

- trigram-based HMM

$$a(uv,vw) \approx P_{MLE}(vw \mid uv) = \frac{c(u,v,w)}{c(u,v)}$$

$$b(O_i,uv) = b(O_i,v) \approx P_{MLE}(O_i \mid v) = \frac{c(v,O_i)}{c(v)}$$

$$\pi(*u) \approx P_{MLE}(*u \mid **) = \frac{c(*,*,u)}{c(**)} \quad \pi(uv) \approx P_{MLE}(uv \mid *u) = \frac{c(*,u,v)}{c(*u)}$$

Exercise

Given the following corpus,

horse/NN flies/NNS time/VBP morning/NN rays/NNS ./.
eat/VB breakfast/NN at/IN morning/NN time/NN ./.
take/VB time/NN with/IN arrow/NN projects/NNS ./.
dinner/NN time/NN goes/VBZ before/IN sleep/NN ./.
flies/NNS smell/VBP an/DT arrow/NN drink/NN ./.
bees/NNS sting/VBP like/IN some/DT flies/NNS ./.

apply MLE to estimate the non-zero parameters for the POS-tags involved in the sentence:

"time flies like horse flies ."

- Using bigrams
- Using trigrams

How is the prob. of a POS-tag sequence computed?

Explanation probability:

Generative model (joint probabilities) instead of conditional model

$$P(X \mid O) = \frac{P(X,O)}{P(O)} \approx P(X,O) \qquad P(O) \; \textit{constant}$$

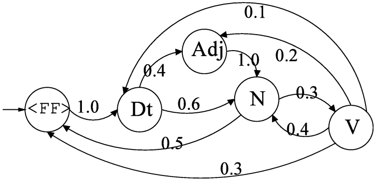
$$P(X_1,..,X_T,O) = P(X_1,..,X_T) \cdot P(O \mid X_1 \ldots X_T)$$

$$P(X_1,..,X_T) = \pi_{X_1} \prod_{t=2}^{T} a_{X_{t-1}X_t}$$
$$P(O \mid X_1 \ldots X_T) = \prod_{t=1}^{T} b_{O_t X_t}$$

$$P(X_1,..,X_T,O) = \pi_{X_1} \cdot b_{O_1 X_1} \cdot \prod_{t=2}^{T} a_{X_{t-1}X_t} \cdot b_{O_t X_t}$$

How is the prob. of a POS-tag sequence computed?

Following the previous example



b_{ik}	.	this	cat	eats	fish	...
<FF>	1.0					
Dt		0.4				
N			0.3		0.3	
V				0.5	0.1	
Adj						

$P(X,O) = P(X, ., \textit{this}, \textit{cat}, \textit{eats}, \textit{fish})$? 8 possible X sequences

X =<FF>,Dt,Adj,N,<FF>
X =<FF>,Dt,Adj,N,V
X =<FF>,Dt,N,<FF>,Dt
X =<FF>,Dt,N,<FF>,Dt
X =<FF>,Dt,N,V,<FF>
X =<FF>,Dt,N,V,N
P(X,O)=$(1 \cdot 1) \cdot (1 \cdot 0.4) \cdot (0.6 \cdot 0.3) \cdot (0.3 \cdot 0.5) \cdot (0.4 \cdot 0.3) = 0.001296$
X =<FF>,Dt,N,V,Adj
X =<FF>,Dt,N,V,Dt

How is the best POS-tag sequence found?

We want to find

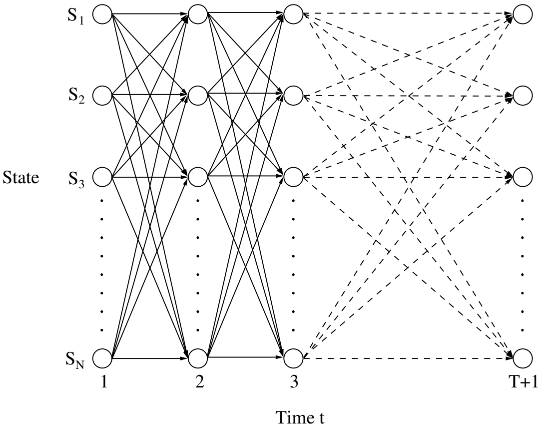
$$\hat{X} = \underset{X}{\operatorname{argmax}} P(X \mid O) \approx \underset{X}{\operatorname{argmax}} P(X, O)$$

- Brute force, $O(N^T)$

  N states (POS tags) and T observations (word sequence length)
- Viterbi algorithm, dinamic programming, $O(T * N^2)$

Outline

Auxiliary structure: Trellis/Lattice



Trellis of a fully connected HMM.

A node $\{s_j, t\}$ of the trellis stores information about states sequences which include $X_t = s_j$.

$$\{s_j, t\} : \quad \delta_t(j) = \max_{X_1,\cdots,X_{t-1}} P(X_1, \cdots, X_{t-1}, s_j, O)$$

$$\varphi_t(j) = last(\operatorname{argmax}_{X_1,\cdots,X_{t-1}} P(X_1, \cdots, X_{t-1}, s_j, O))$$

Viterbi algorithm

1. Initialization: $\forall j = 1 \dots N$

$$\delta_1(j) = \pi_j b_{jo_1}$$

2. Induction: $\forall t = 1 \dots T - 1$

$$\delta_{t+1}(j) = (\max_{1 \leq i \leq N} \delta_t(i) a_{ij}) b_{jo_{t+1}} \quad \forall j = 1 \dots N$$
$$\varphi_{t+1}(j) = \operatorname*{argmax}_{1 \leq i \leq N} \delta_t(i) a_{ij} \quad \forall j = 1 \dots N$$

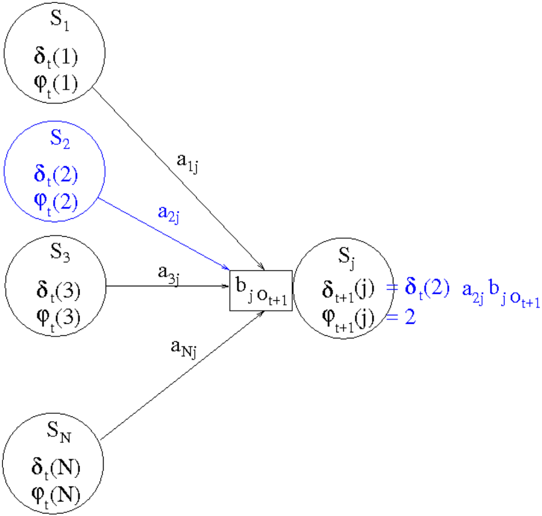
3. Termination

$$\hat{X}_T = \operatorname*{argmax}_{1 \leq i \leq N} \delta_T(i)$$

4. Backwards path readout
   - $\hat{X}_t = \varphi_{t+1}(\hat{X}_{t+1}) \quad \forall t = 1 \dots T - 1$

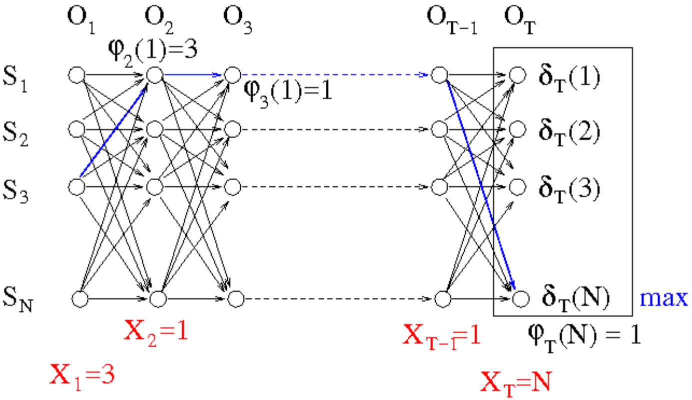
Viterbi algorithm

Induction



Viterbi algorithm

Termination and backwards path readout



Exercise

Apply Viterbi algorithm using the following HMM to

The	kid	fishes	fish
DT	NN	NNS	NN
			NNS
	JJ	VBZ	VBP

A	DT	JJ	NN	NNS	BVZ	VBP
DT		0.2	0.5	0.3		
JJ			0.8	0.2		
NN					1	
NNS						1
VBZ	0.5		0.2	0.3		
VBP	0.4		0.4	0.2		

π	
DT	0.4
JJ	0.2
NN	
NNS	0.3
VBZ	
VBP	0.1

B	the	big	kid	fish	time	fishes	times
DT	1						
JJ		0.8	0.2				
NN			0.3	0.4	0.3		
NNS				0.3		0.4	0.3
VBZ						0.6	0.4
VBP				0.7	0.3		