Syntactic Approaches for Natural Language Processing

Joan Andreu Sánchez

Departamento Sistemas Informáticos y Computación
Instituto Tecnológico de Informática
Universidad Politécnica Valencia

PASCAL 2 Ghana Bootcamp 2011

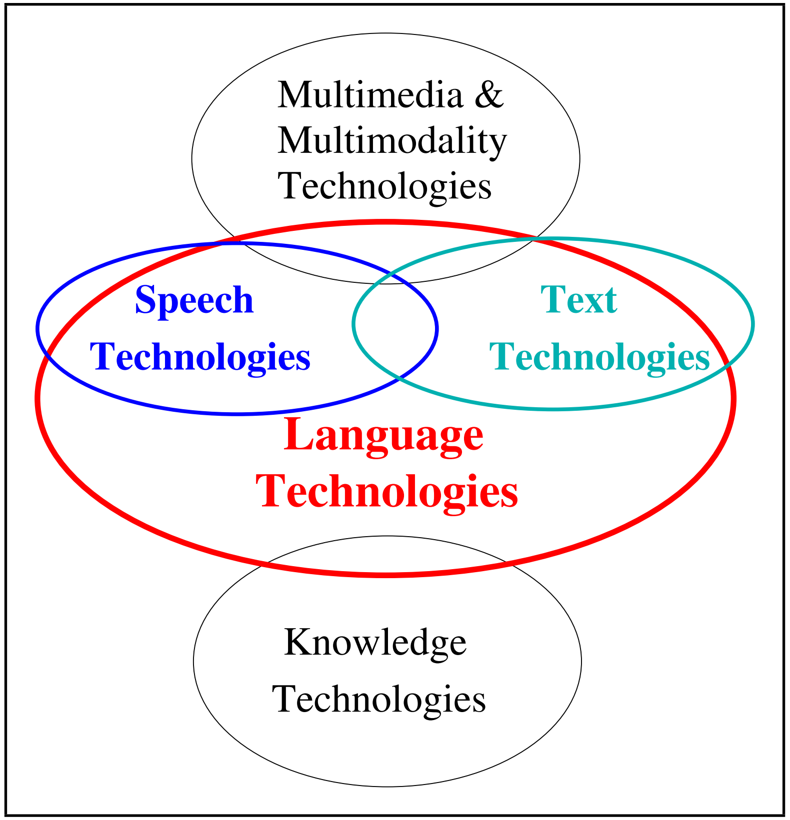
URL: http://www.dsic.upv.es/∼jandreu
e-mail: jandreu@dsic.upv.es

Index

Index

1.1 Introduction

"Computational Linguistics" deals with the most difficult communication process: **Natural Language**



1.1 Introduction

Goal:

To develop systems that are able to process, to understand and, to produce Natural Language

Motivation:

- Natural Language is the main way to represent and to transfer human knowledge
- There exist lots of information and knowledge in Natural Language
- There exist a lot of potential users that need to communicate with computers in Natural Language

1.1 Introduction

Applications

- Systems for information extraction from text and speech

  `Examples:`

  `information retrieval, information extraction, text categorization, ...`

- Systems for speech/text to speech/text:

  `Examples:`

  `machine translation, speech translation, speech recognition, ...`

- Systems for communication with humans:

  `Examples:`

  `dialog systems, query systems, ...`

Probabilistic approach

- Interpretation by using the probabilistic decision rule

  `[ to generate a desired interpretation (output) ]`

- Modeling the human perception with Statistical Decision techniques and Formal Language theory

  `[ to define the statistical dependence between observations (input) and interpretation (output) ]`

- Learning knowledge from examples

  `[ to learn the model parameters from training examples ]`

Main goals of the lecture

- **To introduce syntactic approaches to deal with difficult problems related to Natural Language**

- To study fundamentals related to Computational Linguistics

- To learn basic techniques that are necessary to develop robust systems that are able to understand text data

1.1 Introduction

Applications

- Automatic Speech Recognition
- Machine Translation
- Dialog Systems
- Automatic Summarization
- Text Classification
- Information Retrieval
- . . .

Abstract tasks

- *Language Modeling*
- *Part of Speech Tagging*
- *Parsing*
- Lexical Disambiguation
- Semantic Analysis
- Discourse Analysis
- . . .

Knowledge levels in Natural Language:

- Morphology: word structure
- Syntax:

word category	–	*Part of Speech tagging*
sentence structure	–	*Parsing, Language Modeling*

- Semantics:

  word semantics
  sentence semantics

- Pragmatics: use of the language, cultural issues, environment
- Discourse: dialog structure

1.2 HMM and PoS Tagging

Part of Speech Tagging Problem: Given a set of PoS tags and a sentence, to assign a PoS tag to each word

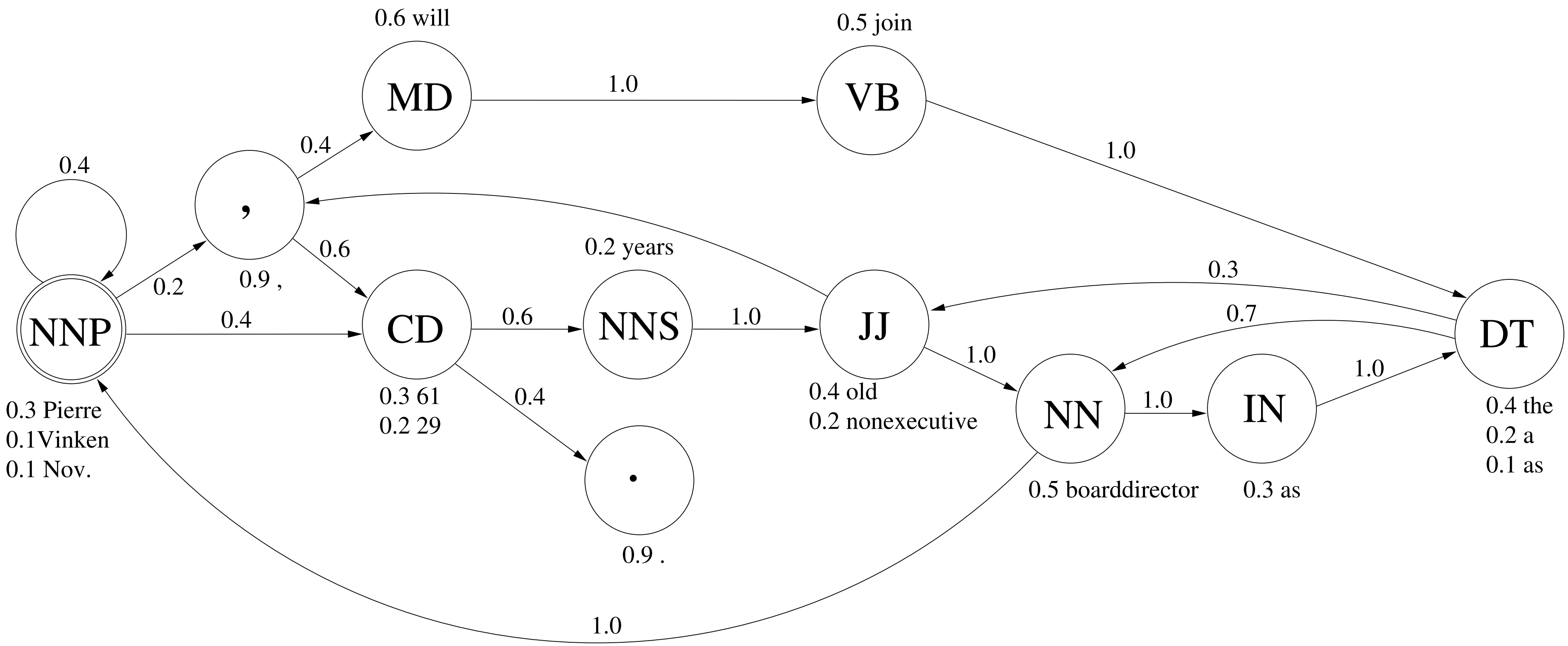
Pierre/NNP Vinken/NNP ,/, 61/CD years/NNS old/JJ ,/, will/MD join/VB the/DT board/NN as/IN a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD ./.

➜ Problem is difficult because of ambiguity

Approaches:

- HMM
- Maximum Entropy
- SVM

HMM for PoS tagging: [Merialdo 94]

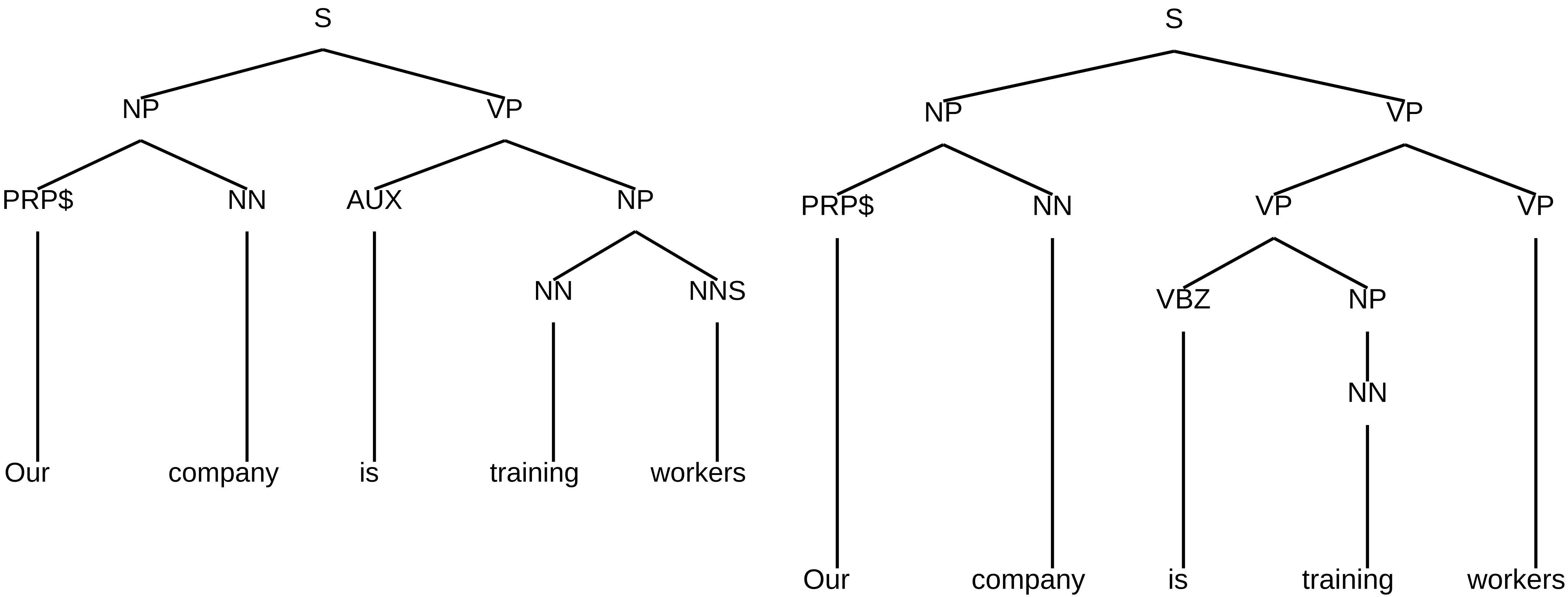


Problems:

- Model learning
- Interpretation

Parsing Problem: Given a sentence, to assign a parsing structure to the sentences

Difficulties in Parsing: Ambiguity

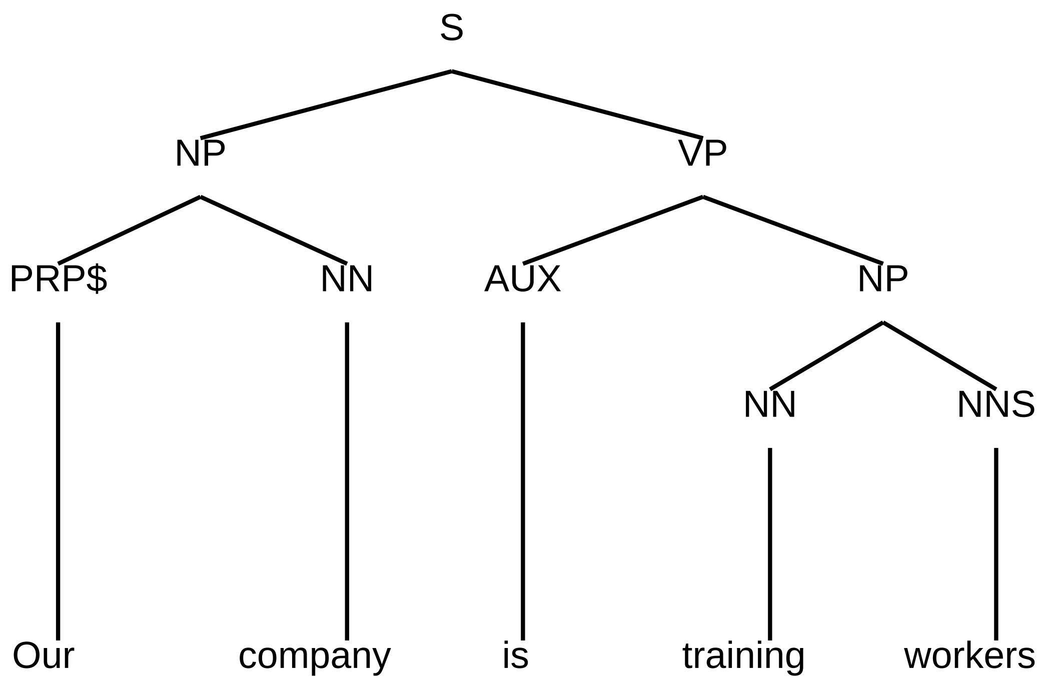


Parsing with syntactic models: (Formal) grammar

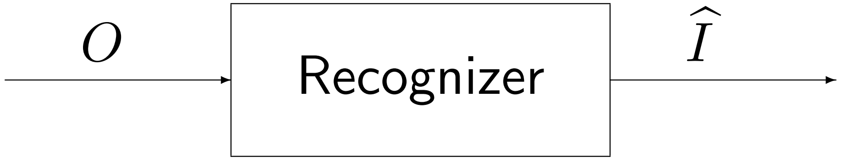
S	$\longrightarrow$	NP VP	PRP$	$\longrightarrow$	Our
NP	$\longrightarrow$	PRP$ NN	NN	$\longrightarrow$	company
NP	$\longrightarrow$	NN NNS	AUX	$\longrightarrow$	is
NP	$\longrightarrow$	NN	NN	$\longrightarrow$	training
VP	$\longrightarrow$	AUX NP	NNS	$\longrightarrow$	workers
VP	$\longrightarrow$	VP VP			
VP	$\longrightarrow$	VBZ NP			

Parsing with syntactic models: (Formal) grammar

1.0	S	→	NP VP	1.0	PRP$	→	Our
0.4	NP	→	PRP$ NN	0.6	NN	→	company
0.3	NP	→	NN NNS	1.0	AUX	→	is
0.3	NP	→	NN	0.4	NN	→	training
0.5	VP	→	AUX NP	1.0	NNS	→	workers
0.3	VP	→	VP VP				
0.2	VP	→	VBZ NP				



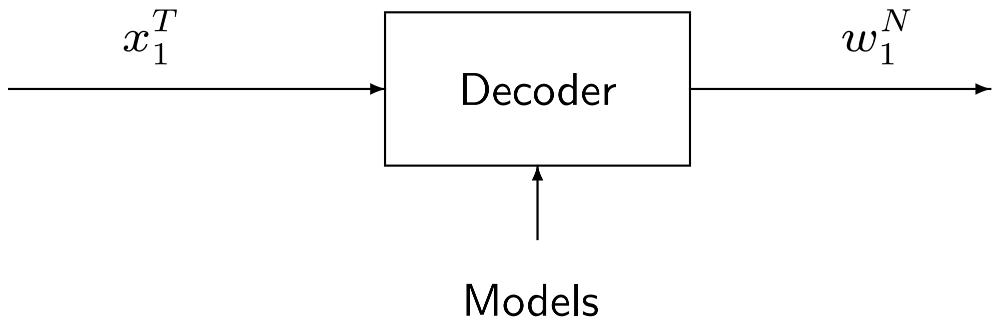
Recognition with noisy channel



$$\widehat{I} = \arg\max_{I} \Pr(I|O) = \arg\max_{I} \Pr(O|I)\Pr(I)$$

$\Pr(I)$: language model probability
$\Pr(O|I)$: channel probability

Automatic Speech Recognition



$$\widehat{w_1^N} = \arg\max_{w_1^N} \Pr(w_1^N | x_1^T) = \arg\max_{w_1^N} \Pr(x_1^T | w_1^N) \Pr(w_1^N)$$

Language Model

$$\Pr(w_1^N) = \Pr(w_1) \prod_{n=2}^{N} \Pr(w_n | w_1^{n-1})$$

1.4 PCFG for Language Modeling

➢ **N-Gram models:** Restriction on the history length w_1^{n-1}

$$\Pr(w_1^N) = \Pr(w_1) \prod_{n=2}^{N} \Pr(w_n | w_{n-k+1}^{n-1})$$

✗ don't capture long-term dependencies
✔ efficient to compute
✔ efficient methods to estimate the model parameters

➢ **Grammatical models:** No restriction on the history length w_1^{n-1}

$$\Pr(w_1^N) = \Pr(w_1) \prod_{n=2}^{N} \Pr(w_n | w_1^{n-1})$$

✔ capture long-term dependencies
✗ expensive to compute
✗ efficient methods to estimate the model parameters, but expensive

Index

Hidden Markov Models: [Vidal 05a, Vidal 05b]

- Simple and compact models for representing regular relations
- Formal framework well understood
- Natural Language is no regular (but almost)
- Adequate representation of short-term syntactic structures
- Adequate modeling of ambiguity

Example

- Primitives: alphabet

  words, punctuation symbols, . . .

- Object representation: written sentences

  "Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29."

- Pattern set

  sentences

- Interpretation: PoS tag association

  Pierre/NNP Vinken/NNP ,/, 61/CD years/NNS old/JJ ,/, will/MD join/VB the/DT board/NN as/IN a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD ./.

2.1 Notation and definitions

- An **alphabet** T is a finite set of symbols.
- A **string** $x = a_1 \cdots a_n \ \ (a_i \in T; i : 1 \ldots n)$, is a finite sequence of symbols of T. The length of the string is noted by $|x|$. Let x and y be two strings, $x, y \in T^*$, then the **concatenation** of x and y is the string xy. $|xy| = |x| + |y|$.
- The **empty string** ϵ, is the string with length equal to zero. For any string x, $x \in T^*$: $\epsilon x = x \epsilon = x$.
- The **closure** T^* is the infinite and countable set of all strings with finite length composed with symbols of T, ϵ included. The **positive closure** T^+ is defined as: $T^+ = T^* - \{\epsilon\}$.
- A **language** L is a set of strings composed with symbols of T $(L \subseteq T^*)$.

2.1 Notation and definitions

A discrete *HMM* is defined as $M = (Q, T, a, b, \pi, q_f)$:

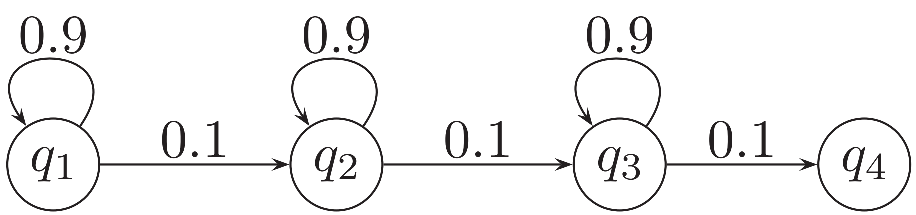
$$a : Q - \{q_f\} \times Q \rightarrow [0,1]; \qquad \forall q \in Q - \{q_f\}\text{: } \textstyle\sum_{q' \in Q} a(q, q') = 1$$

$$b : Q - \{q_f\} \times T \rightarrow [0,1]; \qquad \forall q \in Q - \{q_f\}\text{: } \textstyle\sum_{x \in T} b(q, x) = 1$$

$$\pi : Q \rightarrow [0,1]; \qquad \textstyle\sum_{q \in Q} \pi(q) = 1$$

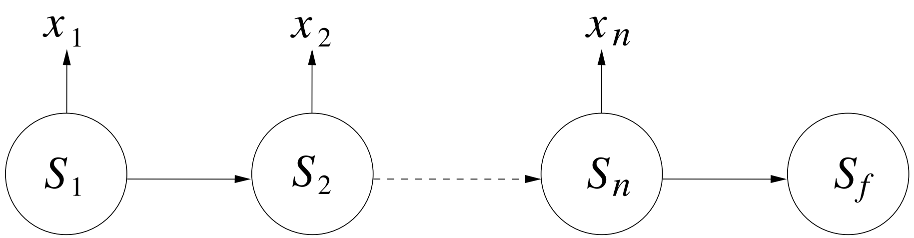
Example: Given $T = \{a, b\}$:

$$\begin{matrix} a \\ b \end{matrix} \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \qquad \begin{matrix} a \\ b \end{matrix} \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix} \qquad \begin{matrix} a \\ b \end{matrix} \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix}$$



$\pi(q_1) = 1.0$

Given $x = x_1 \cdots x_n \in T^*$ and the HMM M:



$$b(s_1 = q_1, x_1)a(s_1, s_2)b(s_2, x_2)\ldots a(s_{n-1}, s_n)b(s_n, x_n)a(s_n, q_f)$$

Let $S = (s_1 = q_1, s_2, \ldots, s_n, s_{n+1} = q_f)$ be a valid path through M. Then:

$$\Pr_M(S) = \prod_{i=1}^{n} a(s_i, s_{i+1}), \quad \text{and} \quad \Pr_M(x \mid S) = \prod_{i=1}^{n} b(s_i, x_i)$$

Let $\mathcal{S}_M(x)$ be the set of all valid paths for x. Then:

$$\Pr_M(x) = \sum_{S \in \mathcal{S}_M(x)} \Pr_M(x \mid S)\Pr_M(S)$$

Forward algorithm

- **Definition:** $\alpha(i,q) = \Pr_M(x_1 \cdots x_i, q) \quad 1 \leq i \leq n+1 \quad q \in Q \cup \{q_f\}$
- **Recursion:** $\forall q \in Q$ with $2 \leq i \leq n$

$$\alpha(i,q) = [\sum_{q' \in Q} \alpha(i-1,q')a(q',q)]b(q,x_i)$$

$$\alpha(n+1,q_f) = \sum_{q' \in Q} \alpha(n,q')a(q',q_f)$$

- **Initialization:** $\alpha(1,q) = \pi(q)b(q,x_1) \qquad \forall q \in Q \cup \{q_f\}$
- **Result:** $\Pr_M(x) = \alpha(n+1,q_f)$

Forward algorithm: Example

	a	b	b	a	
q_1	0.9	0.9 0.9 0.1			
q_2		0.9 0.1 0.9	0.081 0.1 0.9+ 0.081 0.9 0.9		
q_3			0.081 0.1 0.1	0.0729 0.1 0.9+ 0.00081 0.9 0.9	
q_4					0.0072171 0.1

Backward algorithm

– **Definition:** $\beta(i,q) = \Pr_M(x_{i+1} \cdots x_n \mid q) \quad 1 \leq i \leq n+1 \quad q \in Q \cup \{q_f\}$

– **Recursion:** $\forall q \in Q$ with $1 \leq i \leq n-1$:

$$\beta(i,q) = \sum_{q' \in Q} a(q,q') b(q', x_{i+1}) \beta(i+1, q')$$

– **Initialization:** $\beta(n,q) = a(q,q_f)\beta(n+1,q_f) \qquad \forall q \in Q.$
$\beta(n+1,q_f) = 1$

– **Result:** $\Pr_M(x) = b(q_1,x_1)\beta(1,q_1)$

Let:

$$\widehat{S}_x = \max_{S \in \mathcal{S}_M(x)} \Pr_M(x \mid S)\Pr_M(S)$$

and:

$$\widehat{\Pr}_M(x) = \Pr_M(x, \widehat{S}_x)$$

Viterbi algorithm

– **Definition:** $\gamma(i,q) = \widehat{\Pr}_M(x_1 \cdots x_i, q) \quad 1 \leq i \leq n \quad q \in Q \cup \{q_f\}$

– **Recursion:** $\forall q \in Q$ with $2 \leq i \leq n$

$$\gamma(i,q) = [\max_{q' \in Q} \gamma(i-1,q')a(q',q)]b(q,x_i)$$

$$\gamma(n+1,f) = \max_{q' \in Q} \gamma(n,q')a(q',q_f)$$

– **Initialization:** $\gamma(1,q) = \pi(q)b(q,x_1) \qquad \forall q \in Q \cup \{q_f\}$

– **Result:** $\widehat{\Pr}_M(x) = \gamma(n+1,q_f)$

Viterbi algorithm: Example

	a	b	b	a	
q_1	0.9	0.9 0.9 0.1			
q_2		0.9 0.1 0.9	0.081 0.1 0.9 , 0.081 0.9 0.9		
q_3			0.081 0.1 0.1	0.06561 0.1 0.9 , 0.00081 0.9 0.9	
q_4					0.0059049 0.1

Index

3.1 Introduction

- Supervised methods
  - Maximum likelihood estimation

$$\overline{a}(q, q') = \frac{C(q, q')}{C(q)}$$

  - Annotated data is needed
- Non-supervised methods
  - EM algorithms
  - Problem: local optimum

3.1 Introduction

Let M be a HMM and $\theta = (a, b, \pi)$, and let $\Omega = \{x_1, x_2, \ldots, x_n\}$ be a training sample.

$$\widehat{\theta} = \arg\max_{\theta} F_{\theta}(\Omega)$$

- Optimization method
  - Growth transformations
- Optimization function
  - **Maximum likelihood**
  - Corrective training
  - Maximum mutual information

Theorem [Baum 72]

Let $P(\Theta)$ be a homogeneous polynomial with non-negative coefficients. Let $\theta = \{\theta_{ij}\}$ be a point in the domain $D = \{\theta_{ij} \mid \theta_{ij} \geq 0; \sum_{j=1}^{q_i} \theta_{ij} = 1,\ i = 1, \ldots, p; \quad j = 1, \ldots, q_i\}$, and let $Q(\theta)$ be a close transformation in D, that is defined as:

$$Q(\theta)_{ij} = \frac{\theta_{ij}(\partial P/\partial \Theta_{ij})_\theta}{\sum_{k=1}^{q_i} \theta_{ik}(\partial P/\partial \Theta_{ik})_\theta}$$

with the denominator different from zero. Then, $\quad P(Q(\theta)) > P(\theta)$
except if $Q(\theta) = \theta$.

```
input P(Θ)
θ = initial values
repeat
      compute Q(θ) using P(Θ)
      θ = Q(θ)
until convergence
output θ
```

Optimization function

Given a sample Ω and a model M

$$\Pr_M(\Omega, \Delta_\Omega) = \prod_{x \in \Omega} \Pr_M(x, \Delta_M(x)),$$

such that:
– $\Delta_M(x) \subseteq \mathcal{S}_M(x)$
– $\Pr_M(x, \Delta_M(x)) = \sum_{S \in \Delta_M(x)} \Pr_M(x, S)$

– $\forall q, q' \in Q - \{q_f\}$ (See demonstration [Benedí 05])

$$\overline{a}(q, q') = \frac{\sum_{x \in \Omega} \frac{1}{\Pr_M(x, \Delta_M(x))} \sum_{S \in \Delta_M(x)} N((q, q'), S) \Pr_M(x, S)}{\sum_{x \in \Omega} \frac{1}{\Pr_M(x, \Delta_M(x))} \sum_{S \in \Delta_M(x)} N(q, S) \Pr_M(x, S)}$$

– $\forall q \in Q$: $\overline{a}(q, q_f)$
– $\forall q \in Q$, $\forall a \in t$: $\overline{b}(q, a)$

Optimization function

$$\Pr_M(\Omega) = \prod_{x\in\Omega} \Pr_M(x),$$

Baum-Welch algorithm

– $\forall q, q' \in Q - \{q_f\}$

$$\overline{a}(q,q') = \frac{\sum_{x\in\Omega} \frac{1}{\Pr_M(x)} \sum_{i=1}^{n-1} \alpha(i,q)a(q,q')b(q',x_{i+1})\beta(i+1,q')}{\sum_{x\in\Omega} \frac{1}{\Pr_M(x)} \sum_{i=1}^{n} \alpha(i,q)\beta(i,q)}$$

– $\forall q \in Q$: $\overline{a}(q, q_f)$

– $\forall q \in Q$, $\forall a \in t$: $\overline{b}(q, a)$

– $\forall q \in Q$, $\overline{\pi}(q)$

Time complexity: $O(|\Omega||N|b)$

Optimization function

$$\widehat{\Pr}_M(\Omega) = \prod_{x \in \Omega} \widehat{\Pr}_M(x),$$

Viterbi algorithm

– $\forall q, q' \in Q - \{q_f\}$

$$\overline{a}(q, q') = \frac{\sum_{x \in \Omega} N((q, q'), \widehat{S}_x)}{\sum_{x \in \Omega} N(q, \widehat{S}_x)}$$

– $\forall q \in Q$: $\overline{a}(q, q_f)$

– $\forall q \in Q$, $\forall a \in t$: $\overline{b}(q, a)$

– $\forall q \in Q$, $\overline{\pi}(q)$

Time complexity: $O(|\Omega||N|b)$

3.3 Viterbi algorithm

1. Carrying out the maximization with $M^{(i)}$: $\widehat{\mathcal{S}}_x^{(i)}$;

$$\widehat{\mathcal{S}}_x^{(i)} = \{\widehat{S}_x^{(i)} : \widehat{S}_x^{(i)} = \arg \max_{S \in \mathcal{S}_M(x)} \mathrm{Pr}_{M^{(i)}}(x, S)\}$$

2. Applying the transformation: $M^{(i+1)}$.

The function to be optimized is defined after step 1. This function is continous and differentiable:

$$\prod_{x \in \Omega} \mathrm{Pr}_{M^{(i)}}(x, \widehat{S}_x^{(i)}) \leq \prod_{x \in \Omega} \mathrm{Pr}_{M^{(i+1)}}(x, \widehat{S}_x^{(i)}).$$

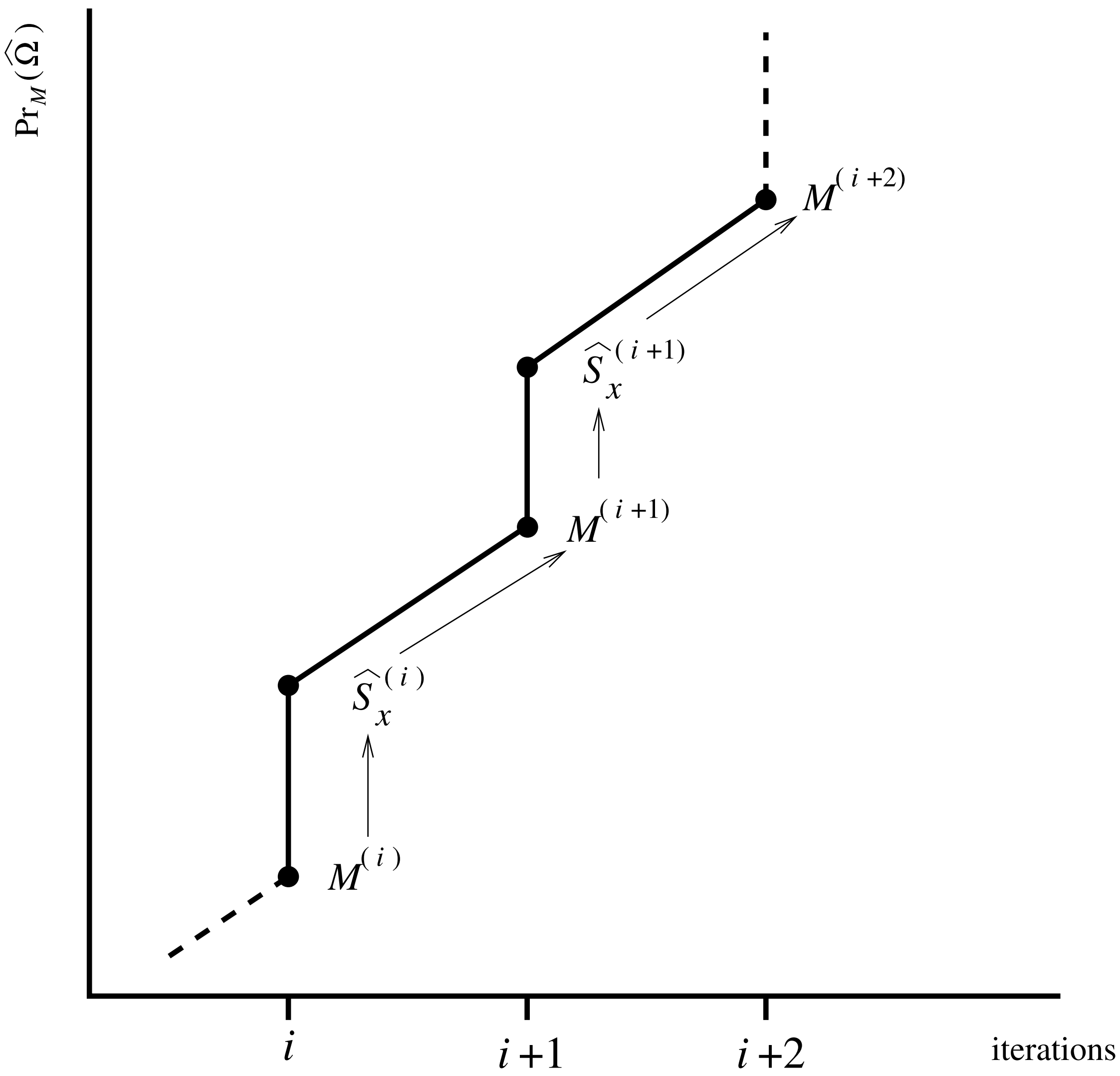
In the next step $i+1$, the most probable sequence $\widehat{S}_x^{(i+1)}$ is computed for each string x with $M^{(i+1)}$, and therefore:

$$\mathrm{Pr}_{M^{(i+1)}}(x, \widehat{S}_x^{(i)}) \leq \mathrm{Pr}_{M^{(i+1)}}(x, \widehat{S}_x^{(i+1)}) \quad \forall x \in \Omega,$$

and hence

$$\prod_{x \in \Omega} \mathrm{Pr}_{M^{(i+1)}}(x, \widehat{S}_x^{(i)}) \leq \prod_{x \in \Omega} \mathrm{Pr}_{M^{(i+1)}}(x, \widehat{S}_x^{(i+1)}).$$

3.3 Viterbi algorithm



3.4 Use of HMM for PoS tagging

Problem: Let W be a sentence and let $\mathcal{C} = \{c_1, c_2, \ldots, c_{\mathcal{C}}\}$ be a PoS tag set:

$$\begin{aligned}\widehat{C} &= \arg \max_{C \in \mathcal{C}^{|W|}} P(c_1 c_2 \ldots c_{|W|} \mid w_1 w_2 w_{|W|}) \\ &= \arg \max_{C \in \mathcal{C}^{|W|}} P(c_1 c_2 \ldots c_{|W|}) P(w_1 w_2 w_{|W|} \mid c_1 c_2 \ldots c_{|W|})\end{aligned}$$

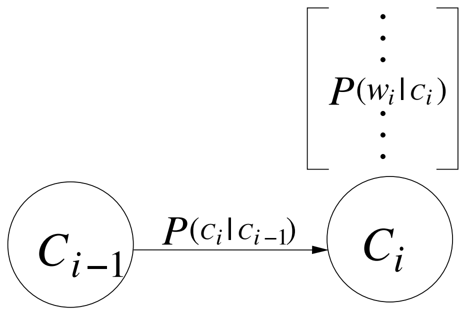
Assumption:

$$\begin{aligned}P(c_1 c_2 \ldots c_{|W|}) &\approx P(c_1) \prod_{i=2}^{|W|} P(c_i | c_{i-1}) \\ P(w_1 w_2 w_{|W|} \mid c_1 c_2 \ldots c_{|W|}) &\approx \prod_{i=1}^{|W|} P(w_i | c_i)\end{aligned}$$

3.4 Use of HMM for PoS tagging

Bigram approach:

$$\widehat{C} = \arg\max_{C \in \mathcal{C}^{|W|}} P(c_1)P(w_1|c_1)\prod_{i=2}^{|W|} P(c_i|c_{i-1})P(w_i|c_i)$$



Problems

- ➢ Labeling: Viterbi algorithm
- ➢ Parameter learning:
  - Non-supervised methods: Baum-Welch estimation.
  - Supervised methods:

$$P(c_i|c_{i-1}) = \frac{f(c_{i-1}c_i)}{f(c_{i-1})} \qquad P(w_i|c_i) = \frac{f(w_i, c_i)}{f(c_i)}$$

Example:

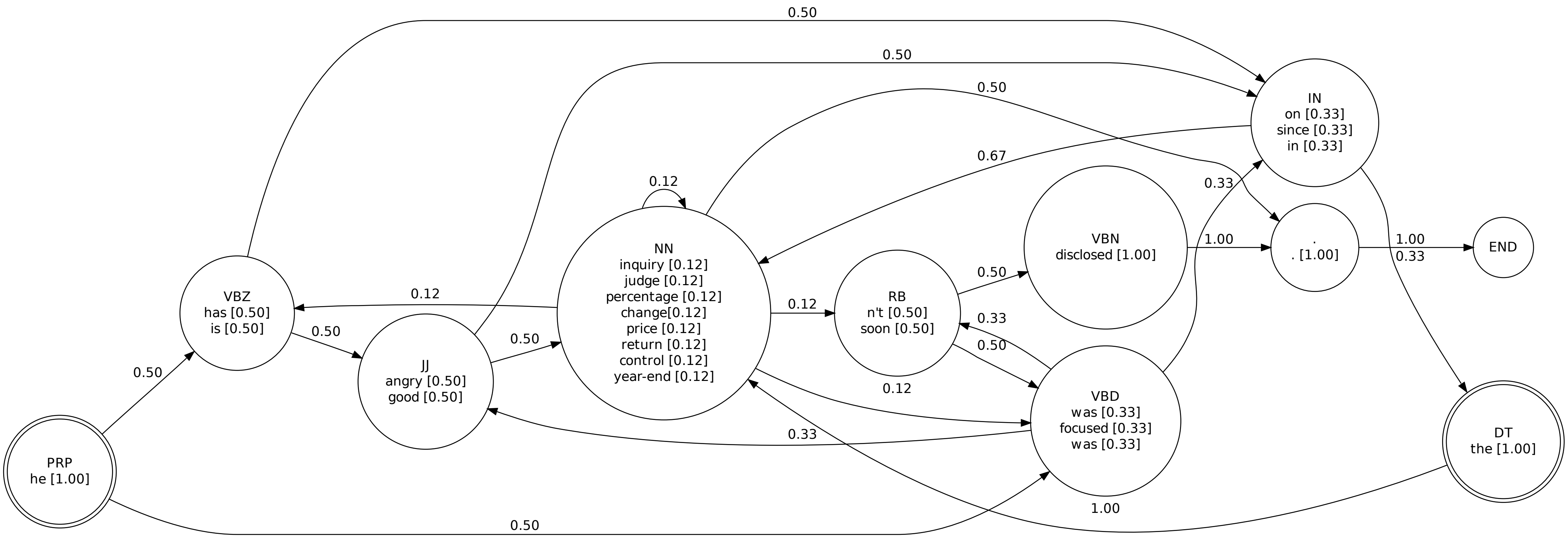
he/PRP has/VBZ good/JJ control/NN ./.

the/DT percentage/NN change/ NN is/VBZ since/IN year-end/ NN ./.

the/DT price/NN was/VBD n't/RB disclosed/VBN ./.

he/PRP becameVBD/ angry/JJ in/IN return/NN ./.

the/DT inquiry/NN soon/RB focused/VBD on/IN the/DT judge/NN ./.



Index

Context-free grammar: [Aho,72]

- Simple and compact models for parsing
- Formal framework well understood
- Adequate representation of long-term syntactic structures
- Adequate modeling of ambiguity

Example

- Primitives: alphabet

  words, punctuation symbols, . . .

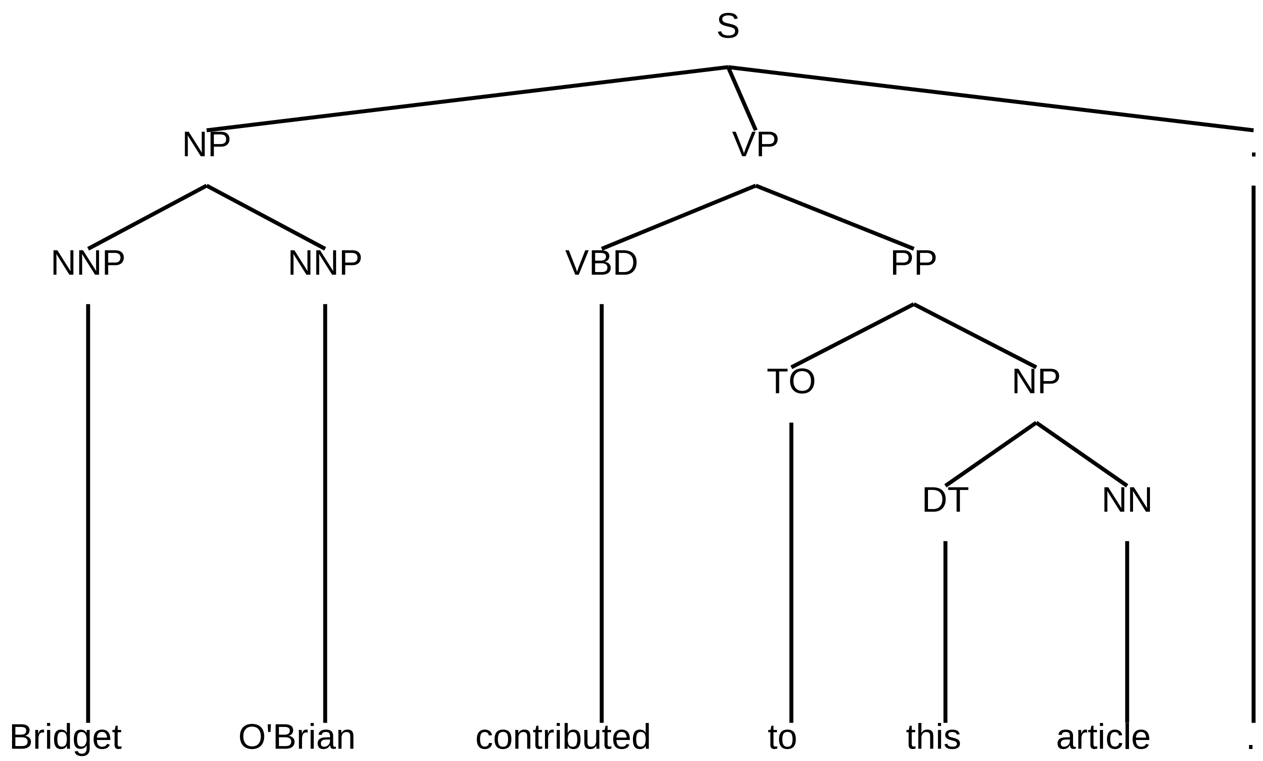
- Object representation: written sentences

  "Bridget O'Brian contributed to this article"

- Pattern set

  sentences

- Interpretation: syntactic analysis



Similar definitions as in HMM:

- **Alphabet**: T is a finite set of symbols.
- **String**: a finite sequence of symbols of T.
- **Closure** T^*: the infinite and countable set of all strings with finite length composed with symbols of T, ϵ included.
- **Language**: L is a set of strings composed with symbols of T $(L \subseteq T^*)$.

4.1 Notation and definitions

➢ **Grammar:** $G = (N, T, P, S)$

$V = N \cup T;\ N \cap T = \emptyset;\ S \in N;\quad (A \rightarrow \beta) \in P;$ $\qquad A \in N; \beta \in V^*$

➢ **Derivation**:

$$\mu A \delta \Longrightarrow \mu \beta \delta \ \underline{\text{iff}}\ \exists (A \rightarrow \beta) \in P; \qquad \mu, \delta \in V^*$$

➢ **Sentential Form**:

$\alpha \in V^*$ is a *sentential form* of G <u>if</u> $S \overset{*}{\Longrightarrow} \alpha$

➢ **Language generated by G**:

$$L(G) = \{x \in T^* \mid S \overset{*}{\Longrightarrow} x\}$$

➢ **Grammar classification**:

- **Type 2**: *context free grammars*
  $A \rightarrow \beta$ $\qquad A \in N; \beta \in V^*$
- **Type 3**: *regular grammars*
  $A \rightarrow aB, \quad A \rightarrow a$ $\qquad A, B \in N; a \in T$

Approaches

- Top-Down parsing
- Down-Top parsing

Depending on time complexity

- Backtracking methods `Exponential complexity`
- Deterministic methods `Linear complexity`

  Grammars: LL(1), SLR(1), LALR(1), LR(1), ...
- Tabular methods `Cubic complexity`

  CKY algorithm
  Earley algorithm [Aho 72, Stolcke 95]

ALGORITHM: Cocke-Kasami-Younger

Input $G = (N, T, P, S)$ in CNF and $\mathbf{x} = x_1 \dots x_n \in T^*$

Output Parsing table $t[i, l] \quad (1 \leq i, l \leq n)$

$A \in t[i, l]$ <u>if</u> $A \stackrel{*}{\Longrightarrow} x_{i+1} \dots x_l$

METHOD

for all $i: \ 0 \dots n-1$ **do**

$$t[i, i+1] := t[i, i+1] \ \cup \ \{A \mid (A \to b) \in P; \ \ b = x_{i+1}\}$$

for all $j: \ 2 \dots n$ **do**

  for all $i: \ 0 \dots n-j$ **do**

    for all $k: \ 1 \dots j-1$ **do**

$$t[i, i+j] := t[i, i+j] \ \cup \ \{A \mid (A \to BC) \in P;$$

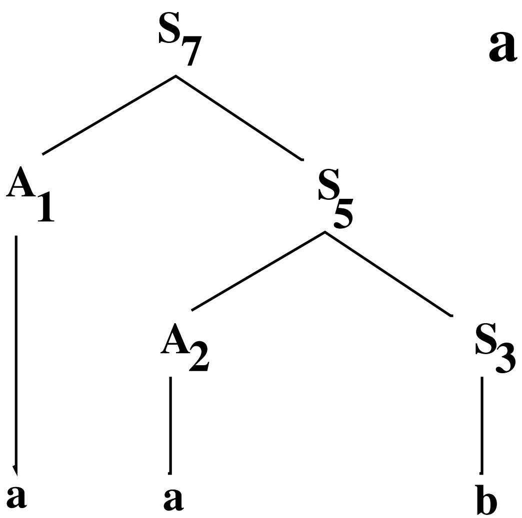
$$B \in t[i, i+k]; \ \ C \in t[i+k, i+j]\}$$

if $S \in t[0, n]$ **then** $x \in L(G)$ **else** $x \notin L(G)$

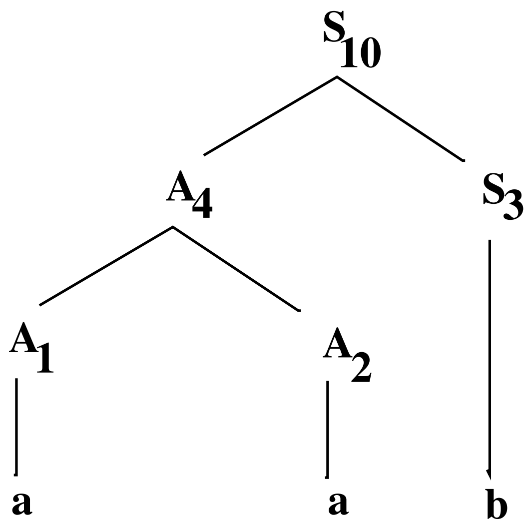
END

S --> A S
S --> b
A --> A S
A --> A A
A --> a

	l = 1	l = 2	l = 3
i = 0	1: (A, 0, 0)	4: (A, 1, 2)	7: (S, 1, 5) 8: (A, 1, 5) 9: (A, 1, 6) 10: (S, 4, 3) 11: (A, 4, 3)
i = 1		2: (A, 0, 0)	5: (S, 2, 3) 6: (A, 2, 3)
i = 2			3: (S, 0, 0)



a a b



4.1 Notation and definitions

Let $x \in T^*$ and a stochastic model M characterized by a parameter vector θ, we are interested in computing: $p_\theta(x)$

Stochastic language (L, ϕ) over T [Wetherell 80]:

- $L \subseteq T^*$ characteristic language
- $\phi : T^* \longrightarrow [0,1]$ computable stochastic function:

  i) $x \notin L \Longrightarrow \phi(x) = 0 \qquad \forall x \in T^*$
  ii) $x \in L \Longrightarrow 0 < \phi(x) \leq 1 \qquad \forall x \in T^*$
  iii) $\sum_{x \in L} \phi(x) = 1$

Example [Booth 73]

Given the alphabet $T = \{a, b\}$, the following language is defined: $L = \{a^n b^n \mid n \geq 0\}$, where $\phi(x) = 0$, $\forall x \notin L$ and $\phi(a^n b^n) = \frac{1}{en!}$

$$\sum_{x \in L} \phi(x) = \sum_{0 \leq n \leq \infty} \frac{1}{en!} = \frac{1}{e} \sum_{0 \leq n \leq \infty} \frac{1}{n!} = \frac{1}{e} e = 1$$

4.1 Notation and definitions

Probabilistic context-free grammar: $G_s = (G, p)$

- $G = (N, T, P, S)$ characteristic grammar

- $p : P \rightarrow]0, 1]$ probability of the rules. $\forall A_i \in N$:

$$\sum_{1 \leq j \leq n_i} p(A_i \rightarrow \alpha_j) = 1,$$

where n_i is the number of rules with A_i in the left side of the rules.

Stochastic derivation for PCFG

Given a sequence of stochastic events:

$$S = \alpha_0 \stackrel{r_1}{\Rightarrow} \alpha_1 \stackrel{r_2}{\Rightarrow} \alpha_2 \cdots \alpha_{m-1} \stackrel{r_m}{\Rightarrow} \alpha_m = x$$

the probability of x being generated by $G_s = (G, p)$ from the rule sequence $d_x = r_1, \ldots, r_m$, is:

$$\Pr_{G_s}(x, d_x) = p(r_1)p(r_2 \mid r_1) \cdots p(r_m \mid r_1 \cdots r_{m-1})$$

- **problem**: computation of the probabilities

- **restriction**: $p(r_j \mid r_1 \cdots r_{j-1}) = p(r_j)$

$$\Pr_{G_s}(x, d_x) = \prod_{j=1 \cdots m} p(r_j)$$

Probability of a derivation $d_x = r_1, \ldots, r_m$

$$\Pr_{G_s}(x, d_x) = \prod_{j=1\cdots m} p(r_j) = \prod_{\forall (A\to\alpha)\in P} p(A \to \alpha)^{\mathrm{N}(A\to\alpha, d_x)}$$

Probability of a string

$$\Pr_{G_s}(x) = \sum_{d_x \in D_x} \Pr_{G_s}(x, d_x)$$

Probability of the best derivation

$$\widehat{\Pr}_{G_s}(x) = \max_{d_x \in D_x} \Pr_{G_s}(x, d_x)$$

Probability of a string with a subset of derivations $\Delta_x \subseteq D_x$

$$\Pr_{G_s}(x, \Delta_x) = \sum_{d_x \in \Delta_x} \Pr_{G_s}(x, d_x)$$

Language generated by a PCFG

$$L(G_s) = \{x \in L(G) \mid \Pr_{G_s}(x) > 0\}$$

Consistent grammar

A PCFG $G_s = (G, p)$ is consistent iff:

$$\sum_{x \in L(G)} \Pr_{G_s}(x) = 1$$

Theorem [Booth 73]

There exist stochastic languages (L, ϕ) that can not be generated by a stochastic grammar $G_s = (G, p)$

Dem. outline Let $L = \{a^n b^n \mid n \geq 0\}$ be a stochastic language:

$$\phi(a^n b^n) = \frac{1}{en!}$$

There is not any G_s such that $\quad \phi(x) = \Pr_{G_s}(x) \quad \forall x \in L$

Inside algorithm for PCFG [Lari 90]

- Given $x = x_1 \ldots x_n \in T^*$ and $A \in N$

$$e(A < i, l >) = \Pr_{G_s}(A \stackrel{*}{\Rightarrow} x_i \ldots x_l)$$

- Compute $\forall A \in N$:

$$e(A < i, i >) = p(A \rightarrow b)\ \delta(b, x_i) \qquad 1 \leq i \leq n$$

$$e(A < i, j >) = \sum_{B,C \in N} p(A \rightarrow BC) \sum_{k=i}^{j-1} e(B < i, k >)\ e(C < k+1, j >) \qquad 1 \leq i < j \leq n$$

- $\Pr_{G_s}(x) = e(S < 1, n >)$
- Time complexity: $O(|x|^3|P|)$

Inside algorithm for PCFG (bracketed version [Pereira 92])

- Bracketed sentence:

  (((Pierre Vinken) , ((61 years) old) ,) (will (join (the board) (as (a nonexecutive director)) (Nov. 29.))) .)

$$c(i,j) = \begin{cases} 1 & \text{if } (i,j) \text{ does not overlap any span in the sentence,} \\ 0 & \text{otherwise.} \end{cases}$$

- Compute $\forall A \in N$:

$$e(A < i,i >) = p(A \rightarrow b)\ \delta(b, x_i) \qquad 1 \leq i \leq n$$

$$e(A < i,j >) = c(i,j) \sum_{B,C \in N} p(A \rightarrow BC) \sum_{k=i}^{j-1} e(B < i,k >)\ e(C < k+1, j >)$$

$$1 \leq i < j \leq n$$

- Linear if full bracketing

Viterbi algorithm for PCFG [Ney 91]

- Given $x = x_1 \ldots x_n \in T^*$ and $A \in N$

$$\widehat{e}(A < i, l >) = \widehat{\Pr}_{G_s}(A \stackrel{*}{\Rightarrow} x_i \ldots x_l)$$

- Compute $\forall A \in N$:

$$\widehat{e}(A < i, i >) = p(A \rightarrow b)\ \delta(b, x_i) \qquad 0 \leq i \leq n$$

$$\widehat{e}(A < i, j >) = \max_{B,C \in N} p(A \rightarrow BC) \max_{k=i,\ldots,j-1} \widehat{e}(B < i, k >)\widehat{e}(C < k+1, j >)$$

$$1 \leq i < j \leq n$$

- $\widehat{\Pr}_{G_s}(x) = \widehat{e}(S < 1, n >)$
- Time complexity: $O(|x|^3|P|)$ (Bracketed version: linear if full bracketing)

Outside algorithm for PCFG

- Given $x = x_1 \ldots x_n \in T^*$ and $A \in N$

$$f(A < i, l >) = \Pr_{G_s}(S \overset{*}{\Rightarrow} x_1 \ldots x_{i-1}\ A\ x_{l+1} \ldots x_n)$$

- Compute $\forall A \in N$:

$$\begin{aligned} f(A < 1, n >) &= \delta(A, S) \\ f(A < i, j >) &= \sum_{B,C \in N} p(B \to CA) \sum_{k=1}^{i-1} f(B < k, j >)\ e(C < k, i-1 >) \\ &+ \sum_{B,C \in N} p(B \to AC) \sum_{k=j+1}^{n} f(B < i, k >)\ e(C < j+1, k >) \end{aligned}$$

$$1 \le i \le j \le n$$

- $\Pr_{G_s}(x) = \sum_{A \in N} f(A < i, i >)\ p(A \to x_i), \qquad 1 \le i \le n$
- Time complexity: $O(|x|^3|P|)$ (Bracketed version: linear if full bracketing)

Probability of an initial substring: LRI algorithm

$$T(A \Rightarrow B) = \sum_{\alpha} \Pr_{G_s}(A \overset{*}{\Rightarrow} B\alpha)$$
$$T(A \Rightarrow BC) = p(A \rightarrow BC) + \sum_{D} T(A \Rightarrow D) p(D \rightarrow BC)$$

- Given $x = x_1 \ldots x_n \in T^*$ and $A \in N$

$$e(A \ll i, l) = \Pr_{G_s}(A \overset{*}{\Rightarrow} x_i \ldots x_l \ldots)$$

- Compute $\forall A \in N$:

$$e(A \ll i, i) = p(A \rightarrow x_i) + \sum_{D} T(A \Rightarrow D)\, p(D \rightarrow x_i) \qquad 1 \leq i \leq n$$
$$e(A \ll i, j) = \sum_{B,C \in N} T(A \Rightarrow BC) \sum_{k=i}^{j-1} e(B < i, k >)\, e(C \ll k+1, j) \qquad 1 \leq i < j$$

- $\Pr_{G_e}(x_1 \ldots x_k \ldots) = e(S \ll 1, k)$

- Time complexity: $O(|x|^3|P|)$

Index

5.1 Introduction

➢ Supervised methods

➢ Maximum likelihood estimation:

➢ Annotated data is needed ("treebank")

$$\widehat{\Pr}(A \rightarrow \alpha) = \frac{C(A \rightarrow \alpha)}{C(A)}$$

➢ Non-supervised methods

➢ EM algorithms

➢ Problem: local optimum

Let G_s a PCFG with parameters θ and a sample $\Omega = \{x_1, x_2, \ldots, x_n\}$.

$$\hat{\theta} = \arg\max_{\theta} F_{\theta}(\Omega)$$

➢ Optimization method: Growth transformations

➢ Optimization function: Maximum likelihood

Theorem [Baum 72]

Let $P(\Theta)$ be a homogeneous polynomial with non-negative coefficients. Let $\theta = \{\theta_{ij}\}$ be a point in the domain $D = \{\theta_{ij} \mid \theta_{ij} \geq 0; \sum_{j=1}^{q_i} \theta_{ij} = 1,\ i = 1, \ldots, p; \quad j = 1, \ldots, q_i\}$, and let $Q(\theta)$ be a close transformation in D, that is defined as:

$$Q(\theta)_{ij} = \frac{\theta_{ij}(\partial P / \partial \Theta_{ij})_\theta}{\sum_{k=1}^{q_i} \theta_{ik}(\partial P / \partial \Theta_{ik})_\theta}$$

with the denominator different from zero. Then, $P(Q(\theta)) > P(\theta)$ except if $Q(\theta) = \theta$.

```
input P(Θ)
θ = initial values
repeat
    compute Q(θ) using P(Θ)
    θ = Q(θ)
until convergence
output θ
```

5.2 Inside-Outside algorithm

Let a PCFG G_s, a sample Ω and a set of derivations Δ_x for each $x \in \Omega$

$$\mathrm{Pr}_{G_s}(\Omega, \Delta_\Omega) = \prod_{x\in\Omega} \mathrm{Pr}_{G_s}(x, \Delta_x)$$

$\forall (A \to \alpha) \in P$ (See demonstration [Benedí 05])

$$\overline{p}(A \to \alpha) = \frac{\sum_{x\in\Omega} \frac{1}{\mathrm{Pr}_{G_s}(x,\Delta_x)} \sum_{\forall d_x \in \Delta_x} \mathrm{N}(A \to \alpha, d_x)\mathrm{Pr}_{G_s}(x, d_x)}{\sum_{x\in\Omega} \frac{1}{\mathrm{Pr}_{G_s}(x,\Delta_x)} \sum_{\forall d_x \in \Delta_x} \mathrm{N}(A, d_x)\mathrm{Pr}_{G_s}(x, d_x)}$$

Optimization function ($\Delta_x = D_x$)

$$\Pr_{G_s}(\Omega) = \prod_{x\in\Omega} \Pr_{G_s}(x)$$

➢ $\forall(A \rightarrow BC) \in P$; y $\forall(A \rightarrow b) \in P$ (See demonstration)

$$\overline{p}(A \rightarrow BC) = \frac{\sum_{x\in\Omega} \frac{p(A\rightarrow BC)}{\Pr_{G_s}(x)} \sum_{i=0}^{n-j} \sum_{j=2}^{n} \sum_{k=1}^{j-1} f(A < i, i+j >) e(B < i, i+k >) e(C < i+k, i+j >)}{\sum_{x\in\Omega} \frac{1}{\Pr_{G_s}(x)} \sum_{i=0}^{n-j} \sum_{j=1}^{n} f(A < i, i+j >) e(A < i, i+j >)}$$

$$\overline{p}(A \rightarrow b) = \frac{\sum_{x\in\Omega} \frac{1}{\Pr_{G_s}(x)} \sum_{i=0,\ b=x_i}^{n-1} f(A < i, i >) p(A \rightarrow b)}{\sum_{x\in\Omega} \frac{1}{\Pr_{G_s}(x)} \sum_{i=0}^{n-j} \sum_{j=1}^{n} f(A < i, i+j >) e(A < i, i+j >)}$$

➢ Time complexity: $O(|LT|^3|P|)$

5.3 Viterbi algorithm

Optimization function ($\Delta_x = \widehat{d}_x$) [Benedí 05]

$$\Pr_{G_s}(\widehat{\Omega}) = \prod_{x \in \Omega} \Pr_{G_s}(x, \widehat{d}_x)$$

➢ $\forall (A \rightarrow \alpha) \in P$

$$\overline{p}(A \rightarrow \alpha) = \frac{\sum_{x \in \Omega} \mathrm{N}(A \rightarrow \alpha, \widehat{d}_x)}{\sum_{x \in \Omega} \mathrm{N}(A, \widehat{d}_x)}.$$

➢ Time complexity: $O(|LT|^3|P|)$

5.4 Probabilistic properties of the estimated PCFG

Theorem [Booth 73] A PCFG is consistent if $\rho(E) < 1$, where $\rho(E)$ is the spectral radius (absolute value of the largest eigenvalue) of matrix E.

Probabilistic expectation matrix: $E = (e_{ij})$, expected number of times that the non-terminal A_j is derived directly from A_i:

$$e_{ij} = \sum_{(A_i \to \alpha)} p(A_i \to \alpha)\mathrm{N}(A_j, \alpha) \qquad\qquad 1 \leq i, j \leq |N|$$

Expectation matrix

$Q = \sum_{i=0}^{\infty} E^i$. If G_s is consistent, then the sum converges to: $Q = (I - E)^{-1}$

Theorem [Sánchez 97] Let $G_s = (G, p)$ be a PCFG and let Ω be a sample from $L(G)$. If $\overline{G}_s = (G, \overline{p})$ is a PCFG obtained from G_s when applying the previous growth transformation, the $\overline{G}_s$ is consistent.

Palindrome language

$$\{ ww^R \mid w \in \{a, b\}^+;\ R = \text{ reverse string}\}$$

➢ Original model

$S \rightarrow AC$ 0.4	$S \rightarrow BB$ 0.1	$C \rightarrow SA$ 1.0	$A \rightarrow a$ 1.0
$S \rightarrow BD$ 0.4	$S \rightarrow AA$ 0.1	$D \rightarrow SB$ 1.0	$B \rightarrow b$ 1.0

➢ Training set: 1000 strings

➢ Initial model to be estimated

- ➢ 5 non-terminals and 2 terminals $\Rightarrow$ 130 rules
- ➢ Random probabilities attached to the rules

Algorithm	kld	Palindromes (%)	Non palindromes (%)
VS	6.00	1.9	98.1
IO	1.88	76.0	24.0

Combination of N-Grams and PCFG for LM [Benedi 05]

$$\Pr(w) = \Pr(w_1 \ldots w_n) = \prod_{k=1}^{n} \Pr(w_k | w_1 \ldots w_{k-1})$$

$$\Pr(w) = \prod_{k=1}^{n} \Pr(w_k | w_{k-n+1} \ldots w_{k-1})$$

$$\boxed{\Pr(w_k | w_1 \ldots w_{k-1}) = \alpha \Pr_N(w_k | w_{k-n+1} \ldots w_{k-1}) + (1 - \alpha) \Pr_{M_s}(w_k | w_1 \ldots w_{k-1})}$$

➩ M_s: a PCFG G_c of categories (PoS tags) and a word-category distribution C_w

$$\Pr_{G_c, C_w}(w_k | w_1 \ldots w_{k-1})$$

WSJ Experiments

➢ WSJ characteristics:

Data set	Directories	No. of senten.	No. of words
Training (full)	00-20	42,075	1,004,073
Training (≤ 50)	00-20	41,315 (98,2%)	959,390 (95,6%)
Tuning	21-22	3,371	80,156
Test	23-24	3,762	89,537

➢ Vocabulary (Training) $10,000$ more frequent words

➢ 3-Gram model: (*linear discounting*)

- *Tuning set perplexity*: 160.3;
- *Test set perplexity*: 167.3;

5.5 Use of PCFG for LM

Test set perplexity

Model	Perplexity		% improvement
	Trigram	Interpolated	
[Chelba 00]	167.1	148.9	10.9
[Roark 01]	167.0	137.3	17.8
IOb	167.3	142.3	14.9

WER

Model	Training Size	Vocabulary Size	LM Weight	WER
[Chelba 00]	20M	20K	16	13.0
[Roark 01]	1M	10K	15	15.1
Treebank trigram	1M	10K	5	16.6
No language model			0	16.8
Current model	1M	10K	6	16.0

Index

Problem definition [Liang 09]:

- Probabilistic model: $p(\mathbf{x}, \mathbf{z}; \theta)$

  Input: $\mathbf{x}$ (a sentence)
  Hidden output: $\mathbf{z}$ (a parse tree)
  Parameters: θ (rule probabilities)

- Given a set of unlabeled example $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$, maxime the marginal log-likelihood:

$$l(\theta) = \sum_{i=1}^{n} \log p(\mathbf{x}^{(i)}; \theta)$$

- Evaluation of the trained model $\widehat{\theta}$: accuracy

$$\text{true output } \mathbf{z}^{(i)} \leftrightarrow \arg\max_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}^{(i)}; \theta)$$

- Training algorithm: EM algorithm [Dempster 77, Neal 98, Cappé 09]

EM algorithm [Liang 09]:

Batch EM

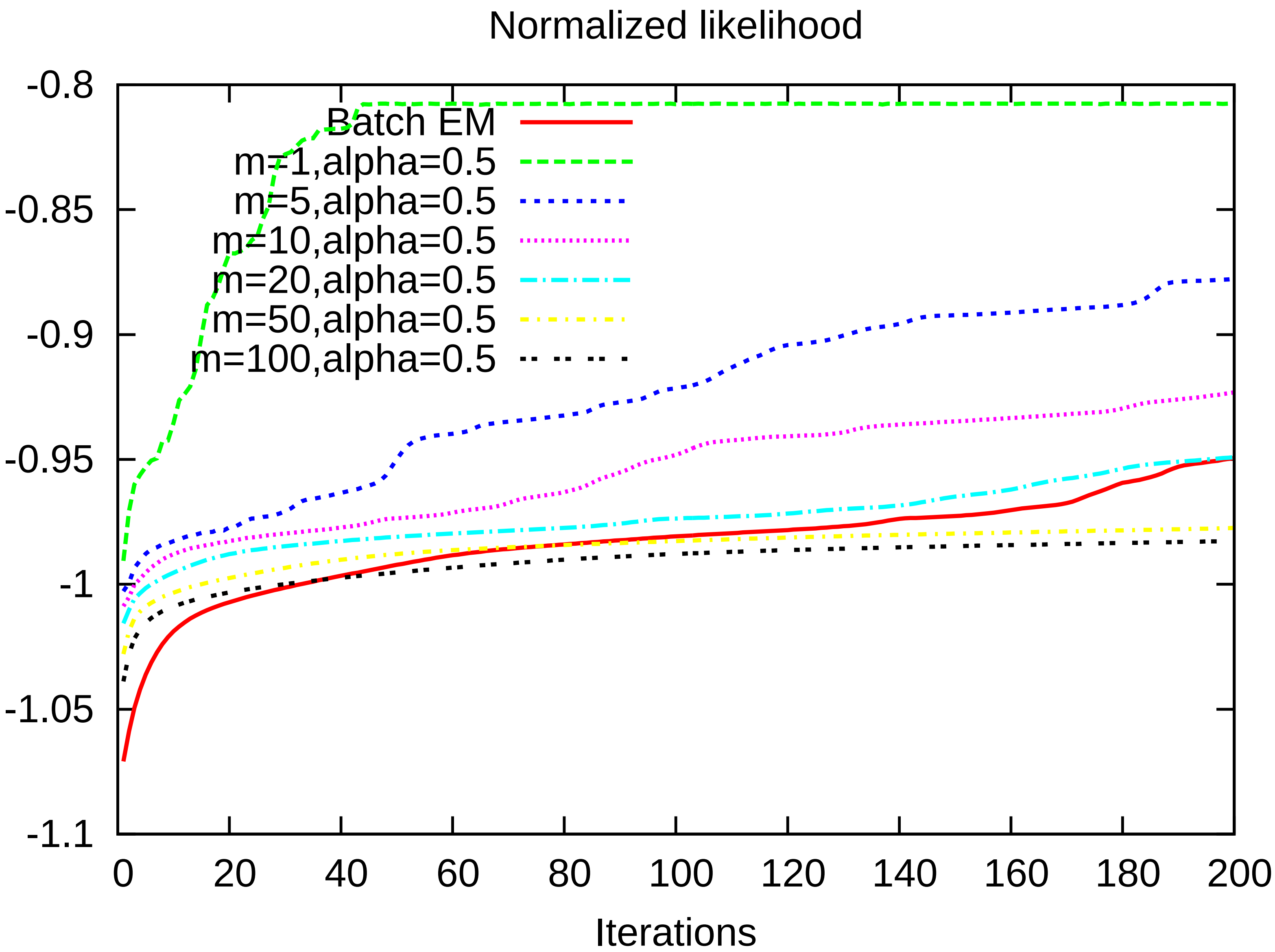
$\mu \leftarrow$ initialization
for each iteration $t = 1, \ldots, T$:
  $\mu' \leftarrow 0$
  for each example $i = 1, \ldots, n$:
    $s'_i \leftarrow \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}^{(i)}; \theta(\mu))\ \phi(\mathbf{x}^{(i)}, \mathbf{z})$
    $\mu' \leftarrow \mu' + s'_i$
$\mu \leftarrow \mu'$

Stepwise EM

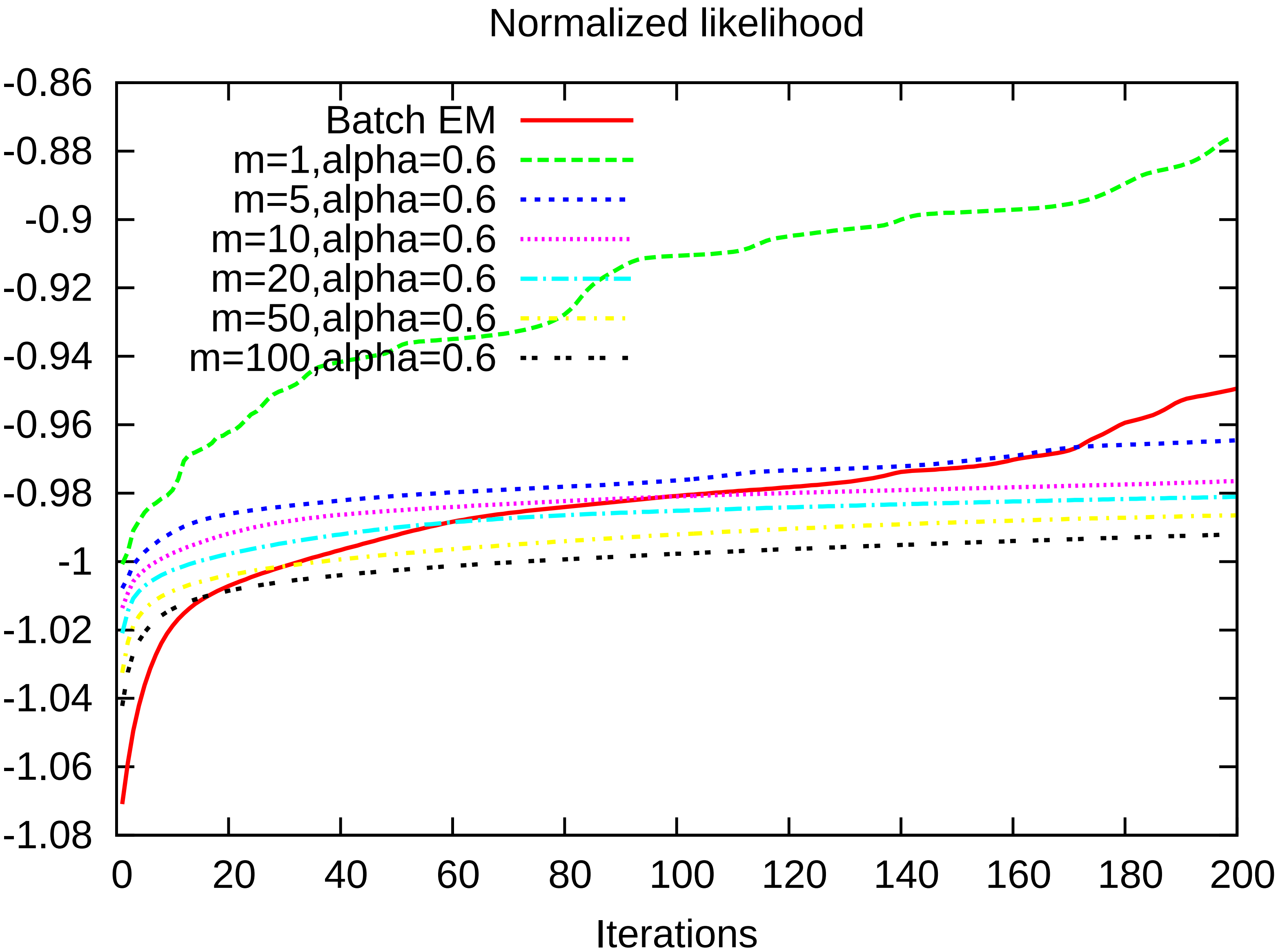
$\mu \leftarrow; k = 0$ initialization
for each iteration $t = 1, \ldots, T$:
  for each example $i = 1, \ldots, n$ in random order:
    $s'_i \leftarrow \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}^{(i)}; \theta(\mu))\ \phi(\mathbf{x}^{(i)}, \mathbf{z})$
    $\mu \leftarrow (1 - \eta_k)\mu + \eta_k s'_i$
    $k \leftarrow k + 1$

- $\phi(\mathbf{x}, \mathbf{z})$: mapping from a labelled example $(\mathbf{x}, \mathbf{z})$ to a vector of sufficient statistics (μ)
- $\theta(\mu)$: maximum likelihood estimate
- Stepwise EM: convergence is guaranteed if $\sum_{k=0}^{\infty} \eta_k = \infty$ and $\sum_{k=0}^{\infty} \eta_k^2 < \infty$
  - $\eta_k = (k+2)^{-\alpha}$ with $0.5 < \alpha \leq 1$
  - Approach: take m examples at once

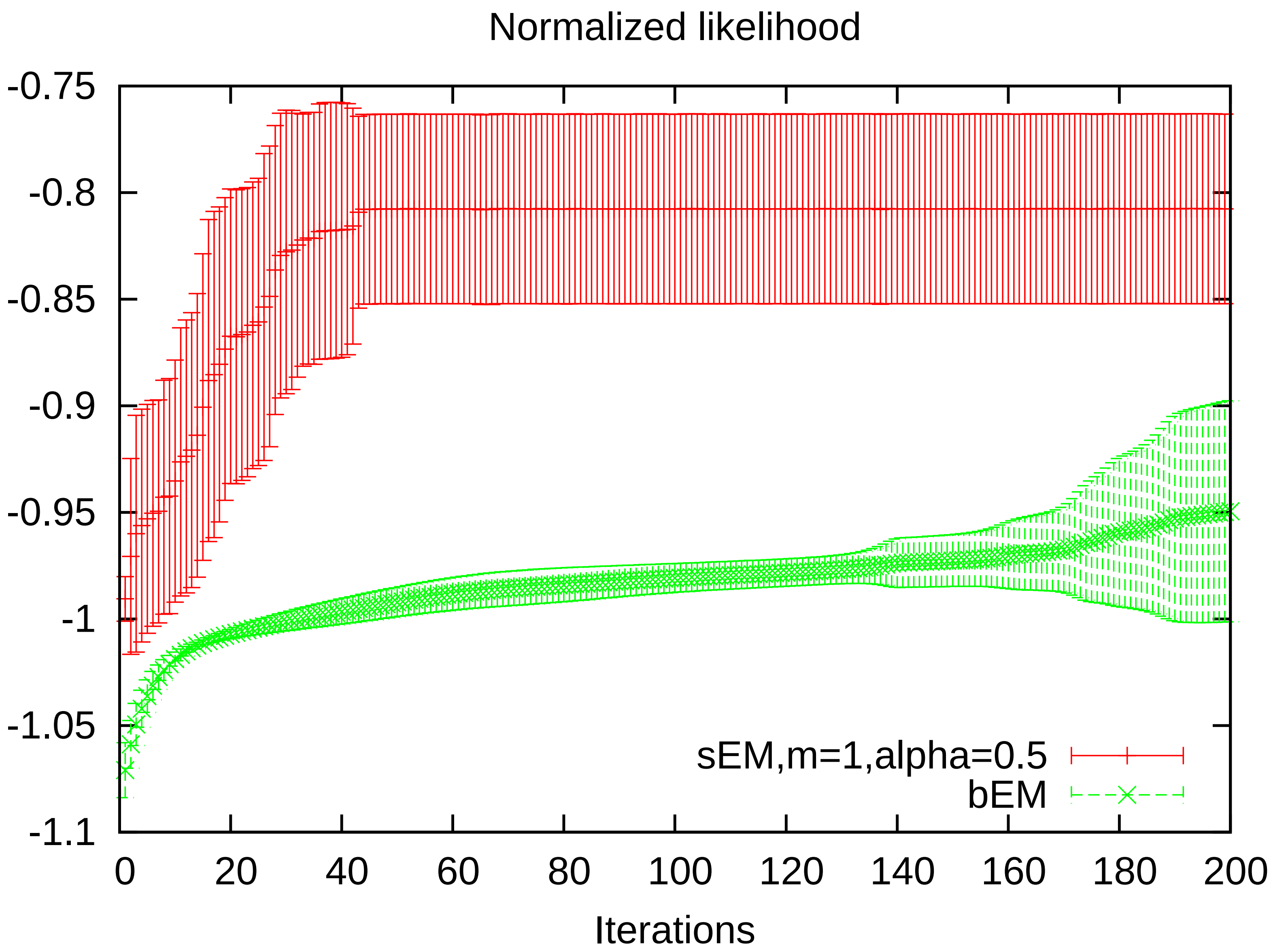
Palindrome language (15 random initializations, $\alpha = 0.5$)



Palindrome language (15 random initializations, $\alpha = 0.6$)

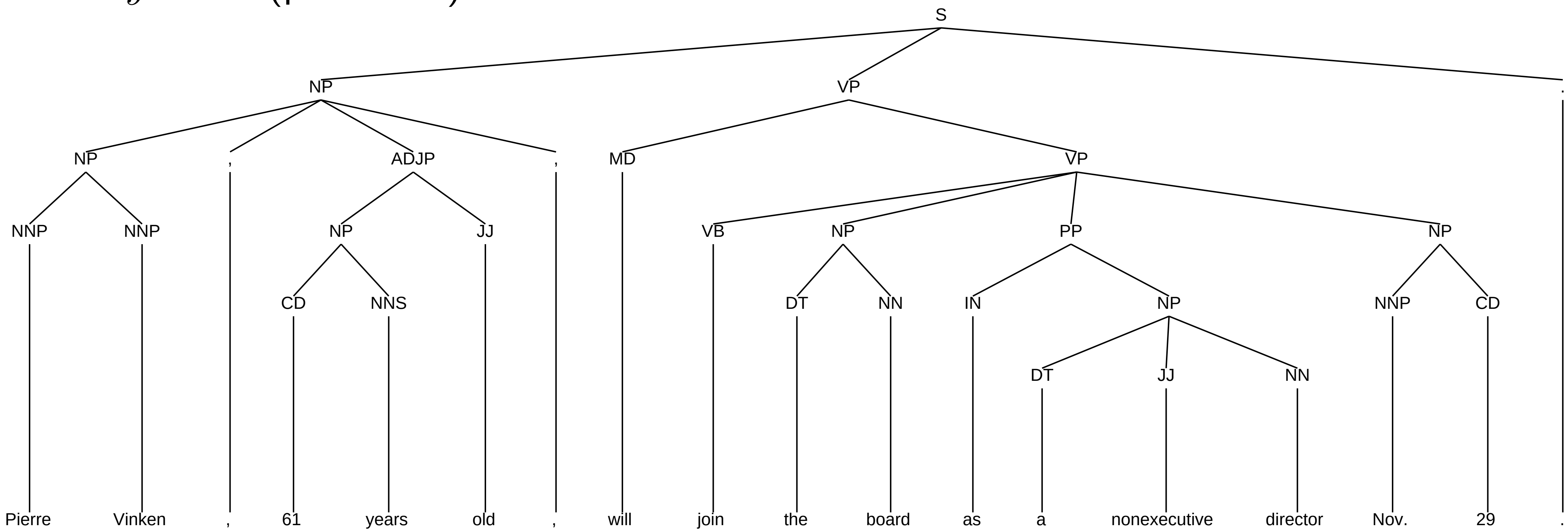


Palindrome language (15 random initializations, $\alpha = 0.5$, confidence interval)



Problem definition:

- Supervised learning: (x, y)
  - x: input data (sentence)
  - y: label (parse tree)



- Problem: to annotate data is slow and expensive
- **Active learning: to annotate just the necessary data**

Pool-based active learning [Settles 08, Settles 10]:

Given: Labeled set $\mathcal{L}$, unlabeled pool $\mathcal{U}$,
query strategy $\phi(\dot{})$, query batch size B
repeat
    // learn a model using the current $\mathcal{L}$
    $\theta = \text{train}(\mathcal{L})$
    for $b = 1$ **to** B **do**
        // query the most informative instance
        $\mathbf{x}_b^* = \arg\max_{\mathbf{x}\in\mathcal{U}} \phi(\mathbf{x})$
        // move the labeled query from $\mathcal{U}$ *to* $\mathcal{L}$
        $\mathcal{L} = \mathcal{L} \cup \langle \mathbf{x}_b^*, \text{label}(\mathbf{x}_b^*) \rangle$
        $\mathcal{U} = \mathcal{U} - \mathbf{x}_b^*$
    end
until *some stopping criterion*

- Similar scheme for parsing in [Hwa 04]

Query strategies:

- **Uncertainty sampling:** to query the instance that is most uncertainty how to label
  - Sequence entropy:

$$\phi^{SE}(\mathbf{x}) = -\sum_{\widehat{\mathbf{y}}} P(\widehat{\mathbf{y}}|\mathbf{x};\theta) \log P(\widehat{\mathbf{y}}|\mathbf{x};\theta)$$

  - Approach: N-best Sequence entropy:

$$\phi^{NSE}(\mathbf{x}) = -\sum_{\widehat{\mathbf{y}} \in \mathcal{N}} P(\widehat{\mathbf{y}}|\mathbf{x};\theta) \log P(\widehat{\mathbf{y}}|\mathbf{x};\theta)$$

- **Information density:** to query the instance that is the most "informative" in average

$$\phi^{ID}(\mathbf{x}) = \phi^{NSE}(\mathbf{x}) \times \left(\frac{1}{U} \sum_{u=1}^{U} \mathsf{sim}(\mathbf{x}, \mathbf{x}^{(u)}) \right)^{\beta}$$

Query strategies for parsing [Hwa 04]:

- Problem space:
  - Based on novelty and frequencies of word pair co-occurrences
  - Based on sentence length: f_{len}
- Performance of the hypothesis:
  - Error-driven function:

$$f_{\text{err}}(\mathbf{w}, G) = 1 - P(\widehat{d}_{\mathbf{w}}|\mathbf{w}, G)$$

  - Normalized tree entropy (similar to $\phi^{SE}(\mathbf{x})$): f_{unc}

Experiments on WSJ UPenn Treebank reported in [Hwa 04]:

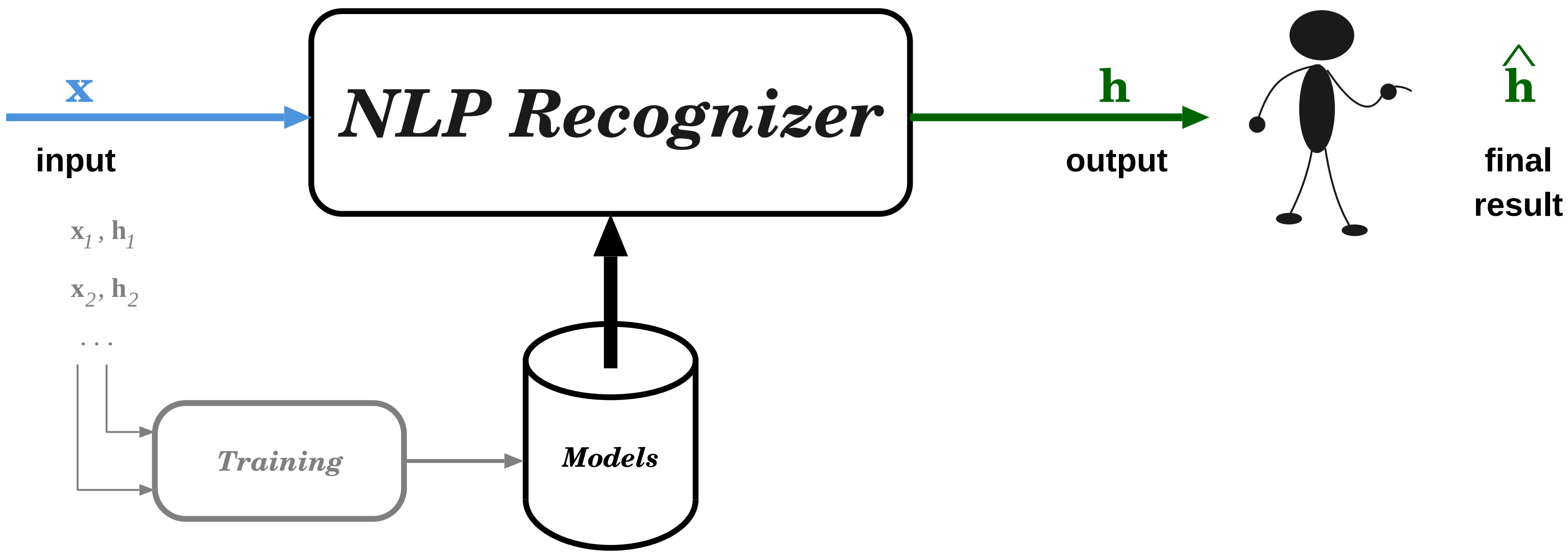
- Collins' model 2 parser
- Learning algorithm: statistics directly over the treebank
- Data:
  - Training: sections 02-21
  - Test: section 23
- Initial model trained on 500 sentences
- Batch size: 100
- Parsing performance: F score

Number of labelled samples at the test performance level of 88%:

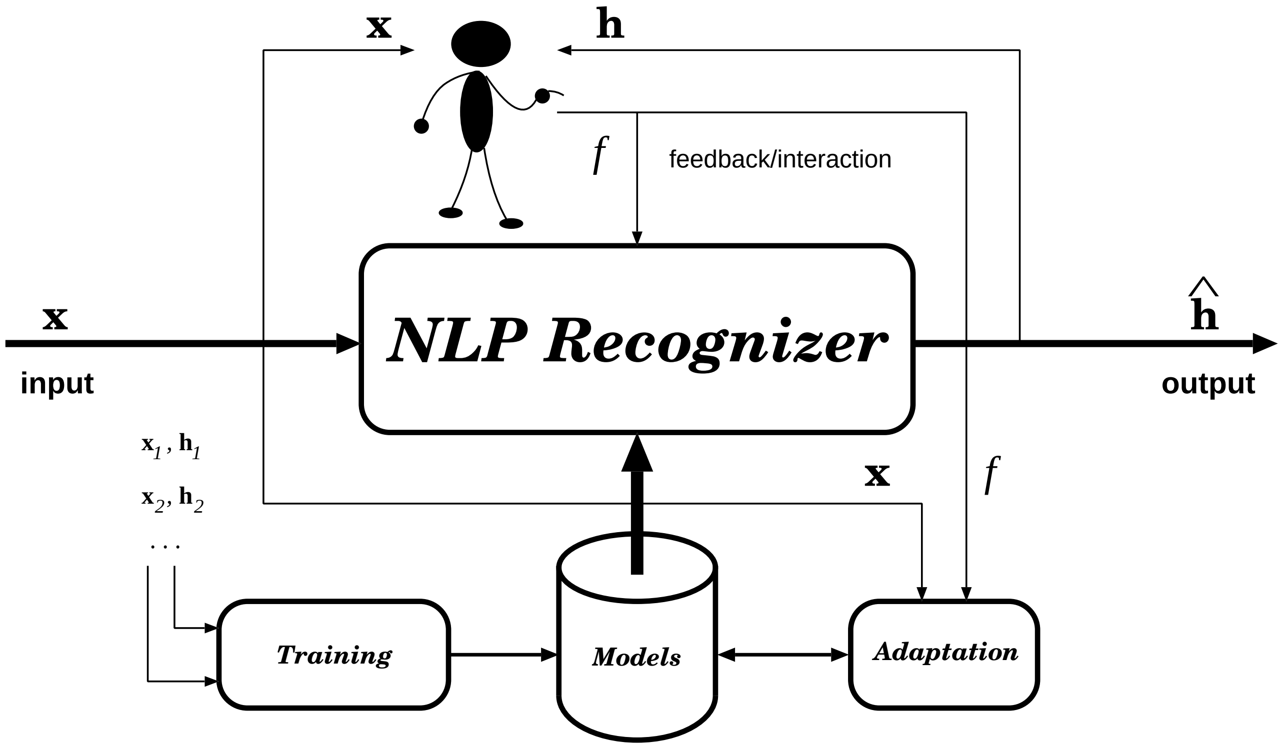
	f_{ran}	f_{len}	f_{err}	f_{unc}
# sentences	$30,500$	-	$20,500$ (33%)	$17,500$ (43%)
# constituents	$695,000$	$625,000$ (10%)	$577,000$ (17%)	$505,000$ (27%)

Problem definition [Sánchez 09, Sánchez 10]: Annotation parse tree is expensive and requires skilled expert humans

- Classical two-step approach:
  1 Apply an automatic system
  2 Manually validate/correct the output



- Interactive Predictive approach:
  - Formally integrate the user into the recognition process
  - The system reacts to user feedback



- New opportunities:
  - Feedback information can be used to create efficient interactive systems
  - Each interaction step yields *ground-truth data*, which allows building *active learning systems*

Classical parsing

$$\widehat{t} = \arg\max_{t \in \mathcal{T}} p_G(t|x)$$

Interactive predictive parsing

$$\widehat{t} = \arg \max_{t \in \mathcal{T} : t_p \in t} p_G(t|x, t_p)$$

$x \rightarrow$ input sentence

$G \rightarrow$ mode (e.g. PCFG)

$\mathcal{T} \rightarrow$ set of all possible trees for x with G

$\widehat{t} \rightarrow$ obtained parse tree

The tree prefix t_p is:

- the corrected constituent, plus
- all its ancestors, plus
- all the constituents to its left

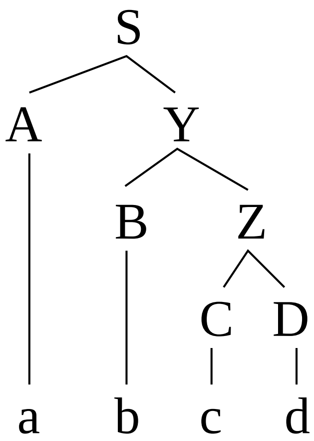
$$t_p(c'^{A}_{ij}) = \{c^B_{mn} : m \leq i, n \geq j, \text{depth}(c^B_{mn}) \leq \text{depth}(c'^{A}_{ij})\} \cup \{c^D_{pq} : p \geq 1, q < i\}$$

IPP parsing

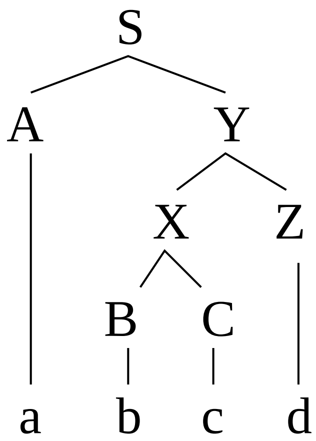
1. The system propose a parse tree $\widehat{t}$
2. The user finds an incorrect constituent c and corrects it, implicitly validating the prefix tree $t_p(c)$
3. The system propose a parse tree $\widehat{t'}$ taking into account the prefix tree $t_p(c)$
4. Go to step 2

n. The user keeps iterating until an error free parse tree is achieved

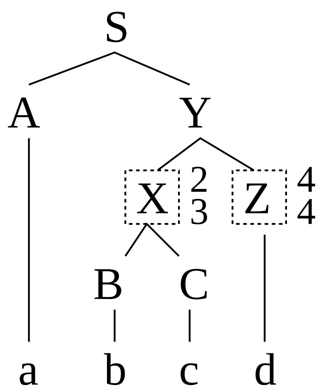
Example:



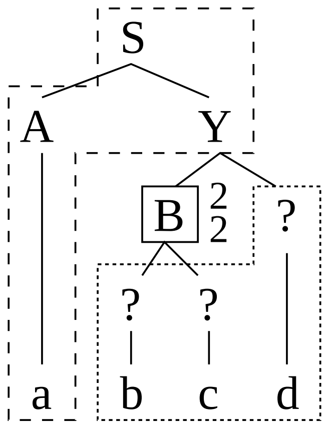
Reference tree



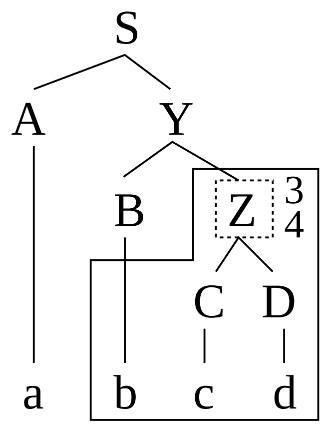
Iteration 0: Proposed output tree 1



Iteration 0: Erroneous constituents



Iteration 1: User corrected constituent



Iteration 1: Proposed output tree 2

Experiments [Sánchez 09]:

- Experiments were performed using the WSJ Treebank and a modified CYK parser
- Vanilla CNF PCFG obtained from sections 02-21. Test set: section 23
- The system simulates user interaction:
  1. Explore the proposed tree and find the first wrong constituent
  2. Replace it with the correct gold constituent
  3. Perform the predictive step (obtain new tree)

  n. Repeat until the gold tree is achieved

Evaluation and results:

- Tree Constituent Error Rate (TCER): Normalized edit distance between the proposed parse tree and the gold tree
  → *User effort when manually postediting the erroneous tree*

- Tree Constituent Action Rate (TCAC): Ration of user constituent corrections performed to obtain the reference tree using the IPP system
  → *User effort when using the IPP system*

PCFG	Baseline		IPP	RelRed
	F_1	TCER	TCAC	
h=0,v=1	0.67	0.40	0.22	45%
h=0,v=2	0.68	0.39	0.21	46%
h=0,v=3	0.70	0.38	0.22	42%

IPP-ANN tool: http://cat.iti.upv.es/ipp/

Parser server

- Custom Viterbi implementation
- Using PCFG in CNF
- Allows requesting subtrees with
  - a root span
  - a complete root constituent

Parser client

- Light Web-client using Flash plugin
- Decodes user feedback
- Requests subtrees to the parse server based on user corrections

Communication

- Client-server communication via sockets
- Using a library specifically tailored for interactive predictive applications

References

References

[Aho 72] A.V. Aho and J.D. Ullman. *The theory of parsing, translation, and compiling. Volumen I: parsing*. Prentice-Hall, 1972.

[Baum 72] L.E. Baum. *An inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes*. Inequalities, 3:1-9, 1972.

[Benedi 05] J.M. Benedí and J.A. Sánchez. *Estimation of stochastic context-free grammars and their use as language models*. Computer Speech and Language, 19(3):249-274, 2005.

[Booth 73] T.L. Booth and R.A. Thompson *Applying Probability Measures to Abstract Languages*. IEEE Transactions on Computers, 22(5):442-450, May 1973.

[Cappé 09] O. Cappé and E. Moulines. *Online Expectation-Maximization Algorithm for Latent Data Models*, Journal of the Royal Statistics Society: Series B (Statistical Methodology), 71, 2009

[Chelba 00] C. Chelba and F. Jelinek. *Structured language modeling*. Computer Speech and Language, 14:283-332, 2000.

[Dempster 77] A.P. Dempster, N.M. Laird and D.B. Rubin. *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society. Series B (Methodological) 39 (1)::1-38, 1977.

[Hwa 04] R. Hwa. *Sample Selection for Statistical Parsing*. Computational Linguistics, 30(3):253-276, 2004.

[Lari 90] K. Lari and S.J. Young. *The Estimation of Stochastic Context-Free Grammars using the Inside-Outside Algorithm*. Computer Speech and Language, 4:35-56, 1990.

References

[Liang 09] P. Liang and D. Klein. *Online EM for Unsupervised Models*. Proc. 10th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT), June, 2009, 611-619.

[Maryanski 79] F.J. Maryanski and M.T. Thomason. *Properties of stochastic syntax-directed tranlation schemata*. *Journal of Computer and Information Sciences*, 8(2):89-110, 1979.

[Merialdo 94] B. Merialdo. *Tagging English Text with a Probabilistic Model*. Computational Linguistics, 20(2):155-171, 1994.

[Neal 98] R. Neal and G. Hinton *A view of the EM algorithm that justifies incremental, sparse, and other variants*. Learning in Graphical Models, 355-368, 1999.

[Ney 91] H. Ney. *Dynamic Programing Parsing for Context-Free Grammars in Continous Speech Recognition*. IEEE Trans. Signal Processing, 39(2):336-340, 1991.

[Pereira 92] F. Pereira and Y. Schabes. *Inside-outside reestimation from partially bracketed corpora*. Proceedings of the 30th Annual Meeting of the ACL, 128-135, 1992.

[Roark 01] B. Roark. *Probabilistic Top-Down Parsing and Language Modeling*. Computational Linguistics, 27(2):249-276, 2001.

[Sánchez 97] J.A. Sánchez and J.M. Benedí. *Consistency of Stochastic Context-Free Grammars from Probabilistic Estimation Based on Growth Transformation*. IEEE Trans. Pattern Analysis and Machine Intelligence, 19(2):1052-1055, 1997.

References

[Sánchez 09] R. Sánchez-Sáez, J.A. Sánchez and J.M. Benedí. *Interactive predictive parsing.* In Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09), 222-225, Paris, France, 2009.

[Sánchez 10a] R. Sánchez-Sáez, L. Leiva, J.A. Sánchez and J.M. Benedí. *Interactive Predictive Parsing using a Web-based Architecture*. Proceedings of the NAACL HLT 2010 Demonstration Session, 37-40, Los Angeles, California, 2010.

[Settles 08] B. Settles and M. Craven. *An Analysis of Active Learning Strategies for Sequence Labelling Tasks*, Empirical Methods in Natural Language Processing (EMNLP), 1069-1078, 2008.

[Settles 10] B. Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2010.

[Stolcke 95] A. Stolcke. *An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities*. Computational Linguistics, 21(2):165-200, 1995.

[Vidal 05a] E. Vidal and F. Thollard and C. de la Higuera, F. Casacuberta and R. Carrasco *Probabilistic finite-state machines - Part I*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(7):1013-1025, 2005.

[Vidal 05b] E. Vidal and F. Thollard and C. de la Higuera, F. Casacuberta and R. Carrasco *Probabilistic finite-state machines - Part II*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(7):1025-1039, 2005.

[Wetherell 80] C.S. Wetherell. *Probabilistic Languages: A Review and some Open Questions*. Computing Surveys, 12(4):361-379, 1980.

Appendices

A growth transformation can be defined as:

$$\overline{p}(A \to \alpha) = \frac{p(A \to \alpha)\left(\frac{\partial \mathrm{Pr}_{G_s}(\Omega,\Delta_\Omega)}{\partial p(A\to\alpha)}\right)_p}{\sum_{i=1}^{n_A} p(A \to \alpha_i)\left(\frac{\partial \mathrm{Pr}_{G_s}(\Omega,\Delta_\Omega)}{\partial p(A\to\alpha_i)}\right)_p}$$

Numerator:

$$\begin{aligned} p(A \to \alpha)\left(\frac{\partial \mathrm{Pr}_{G_s}(\Omega,\Delta_\Omega)}{\partial p(A\to\alpha)}\right)_p &= \mathrm{Pr}_{G_s}(\Omega,\Delta_\Omega)\sum_{x\in\Omega}\frac{p(A\to\alpha)}{\mathrm{Pr}_{G_s}(x,\Delta_x)}\left(\frac{\partial \mathrm{Pr}_{G_s}(x,\Delta_x)}{\partial p(A\to\alpha)}\right)_p \\ &= \mathrm{Pr}_{G_s}(\Omega,\Delta_\Omega)\sum_{x\in\Omega}\frac{p(A\to\alpha)}{\mathrm{Pr}_{G_s}(x,\Delta_x)}\sum_{\forall d_x\in\Delta_x}\left(\frac{\partial \mathrm{Pr}_{G_s}(x,d_x)}{\partial p(A\to\alpha)}\right)_p \\ &= \mathrm{Pr}_{G_s}(\Omega,\Delta_\Omega)\sum_{x\in\Omega}\frac{1}{\mathrm{Pr}_{G_s}(x,\Delta_x)}\sum_{\forall d_x\in\Delta_x}\mathrm{N}(A\to\alpha,d_x)\mathrm{Pr}_{G_s}(x,d_x) \end{aligned}$$

Denominator:

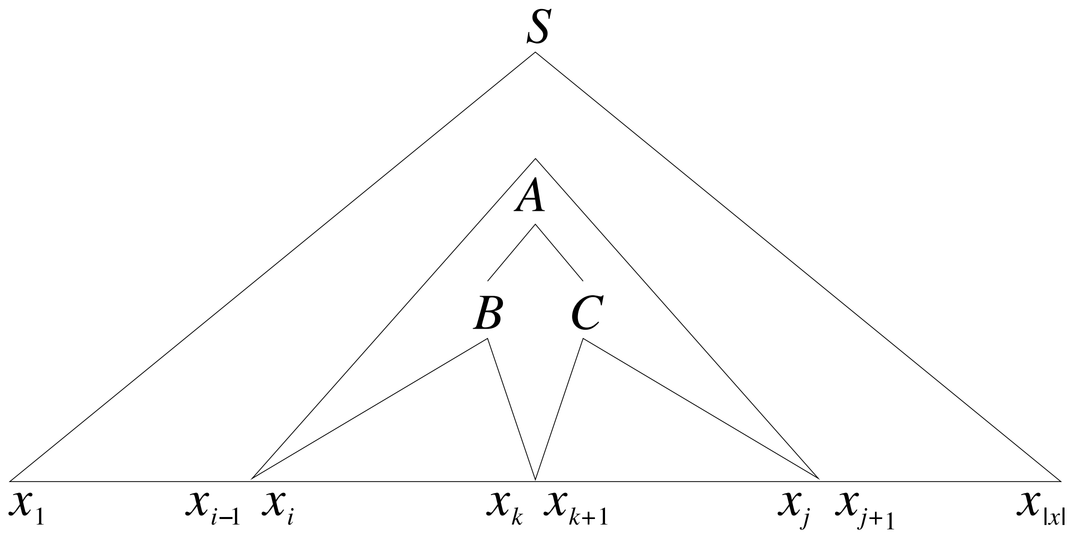
$$\sum_{i=1}^{n_A} p(A \rightarrow \alpha_i) \left(\frac{\partial \mathrm{Pr}_{G_s}(\Omega, \Delta_\Omega)}{\partial p(A \rightarrow \alpha_i)} \right)_p =$$

$$= \quad \mathrm{Pr}_{G_s}(\Omega, \Delta_\Omega) \sum_{x \in \Omega} \frac{1}{\mathrm{Pr}_{G_s}(x, \Delta_x)} \sum_{\forall d_x \in \Delta_x} \sum_{i=1}^{n_A} \mathrm{N}(A \rightarrow \alpha_i, d_x) \mathrm{Pr}_{G_s}(x, d_x)$$

$$= \quad \mathrm{Pr}_{G_s}(\Omega, \Delta_\Omega) \sum_{x \in \Omega} \frac{1}{\mathrm{Pr}_{G_s}(x, \Delta_x)} \sum_{\forall d_x \in \Delta_x} \mathrm{N}(A, d_x) \mathrm{Pr}_{G_s}(x, d_x).$$

Appendix B

- Let $A \to BC$ in a position delimited by integers i, j, k, $1 \leq i \leq k < j \leq |x|$



- $\Delta_{x,i,j,k,A \to BC} \subseteq D_x$: subset of derivations of x in which the rule $A \to BC$ appears delimited by positions i, j, k
- $\Delta_{x,i,j,A}$: subset of derivations of x in which the non-terminal A appears delimited by positions i, j

Appendix B

- $$\sum_{\forall d_x \in D_x} \mathrm{N}(A \rightarrow BC, d_x) \Pr_{G_s}(x, d_x) = \sum_{1 \leq i \leq k < j \leq |x|} \sum_{\forall d_x \in \Delta_{x,i,j,k,A \rightarrow BC}} \Pr_{G_s}(x, d_x)$$
$$= \sum_{1 \leq i \leq k < j \leq |x|} \Pr_{G_s}(S \stackrel{*}{\Rightarrow} x_1 \ldots x_{i-1} A x_{j+1} \ldots x_{|x|}) \cdot$$
$$p(A \rightarrow BC) \cdot \Pr_{G_s}(B \stackrel{*}{\Rightarrow} x_i \ldots x_k) \cdot \Pr_{G_s}(C \stackrel{*}{\Rightarrow} x_{k+1} \ldots x_j)$$
$$= \sum_{1 \leq i \leq k < j \leq |x|} f(A < i, j >) p(A \rightarrow BC) e(B < i, k >) e(C < k+1, j >),$$

- $$\sum_{\forall d_x \in D_x} \mathrm{N}(A, d_x) \Pr_{G_s}(x, d_x) = \sum_{1 \leq i \leq j \leq |x|} \sum_{\forall d_x \in \Delta_{x,i,j,A}} \Pr_{G_s}(x, d_x)$$
$$= \sum_{1 \leq i \leq j \leq |x|} \Pr_{G_s}(S \stackrel{*}{\Rightarrow} x_1 \ldots x_{i-1} A x_{j+1} \ldots x_{|x|}) \Pr_{G_s}(A \stackrel{*}{\Rightarrow} x_i \ldots x_j)$$
$$= \sum_{1 \leq i \leq j \leq |x|} f(A < i, j >) e(A < i, j >).$$

EM algorithm [Neal 98]:

E step: Compute a distribution $\widetilde{p}^{(t)}$ over the range of $\mathbf{Z}$ such that

$$\widetilde{p}^{(t)}(\mathbf{z}) = p(\mathbf{z}|\mathbf{x};\theta^{(t-1)})$$

M step: Set $\theta^{(t)}$ to the θ that maximizes $E_{\widetilde{p}^{(t)}}[\log p(\mathbf{x},\mathbf{z};\theta)]$