Probabilistic Linguistics

Matilde Marcolli
CS101: Mathematical and Computational Linguistics

Winter 2015

Bernoulli measures

- finite set $\mathfrak{A}$ alphabet, strings of arbitrary (finite) length $\mathfrak{A}^\star = \cup_m \mathfrak{A}^m$
- Alphabet: letters, phonemes, lexical list of words,...
- $\Lambda_{\mathfrak{A}}$ = infinite strings in alphabet $\mathfrak{A}$; cylinder sets $\Lambda_{\mathfrak{A}}(w) = \{\alpha \in \Lambda_{\mathfrak{A}} \,|\, \alpha_i = w_i,\ i = 1, \dots, m\}$ with $w \in \mathfrak{A}^m$ (also called the ω-language $\mathfrak{A}^\omega$)
- $\Lambda_{\mathfrak{A}}$ Cantor set with the topology generated by cylinder sets

- Bernoulli measure: $P = (p_a)_{a\in\mathfrak{A}}$ probability measure $p_a \geq 0$ for all $a \in \mathfrak{A}$ and $\sum_{a\in\mathfrak{A}} p_a = 1$
- Gives measure μ_P on $\Lambda_{\mathfrak{A}}$ with $\mu_P(\Lambda_{\mathfrak{A}}(w)) = p_{w_1} \cdots p_{w_m}$
- meaning: in a word $w = w_1 \cdots w_m$ each letter $w_i \in \mathfrak{A}$ is an independent random variable drawn with probabilities $P = (p_a)_{a\in\mathfrak{A}}$

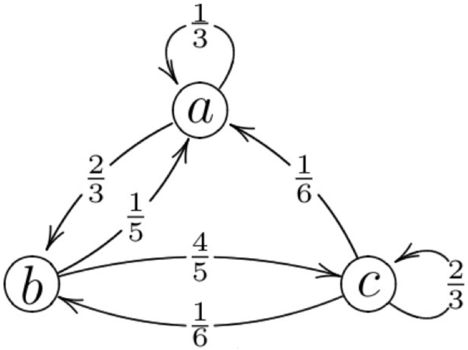
Markov measures

- same as above: $\mathfrak{A}$ alphabet, $\mathfrak{A}^\star$ finite strings, $\Lambda_{\mathfrak{A}}$ infinite strings
- $\pi = (\pi_a)_{a\in\mathfrak{A}}$ probability distribution $\pi_a \geq 0$ and $\sum_a \pi_a = 1$
- $P = (p_{ab})_{a,b\in\mathfrak{A}}$ stochastic matrix $p_{ab} \geq 0$ and $\sum_a p_{ab} = 1$
- Perron–Frobenius eigenvector $\pi P = \pi$
- Markov measure $\mu_{\pi,P}$ on $\Lambda_{\mathfrak{A}}$ with

$\mu_{\pi,P}(\Lambda_{\mathfrak{A}}(w)) = \pi_{w_1} p_{w_1 w_2} \cdots p_{w_{m-1} w_m}$

- meaning: in a word $w_1 \cdots w_m$ letters follow one another according to a Markov chain model, with probability p_{ab} of having a and b as consecutive letters

Example of Markov Chain



• support of Markov measure $\mu_{\pi,P}$ subset $\Lambda_{A,\mathfrak{A}} \subset \Lambda_{\mathfrak{A}}$ with $A = (A_{ab})$ entries $A_{ab} = 0$ if $p_{ab} = 0$ and $A_{ab} = 1$ if $p_{ab} \neq 0$

$$\Lambda_{A,\mathfrak{A}} = \{\alpha \in \Lambda_{\mathfrak{A}} \,|\, A_{\alpha_i \alpha_{i+1}} = 1, \forall i\}$$

• both Bernoulli and Markov measures are shift invariant

$$\sigma : \Lambda_{\mathfrak{A}} \to \Lambda_{\mathfrak{A}}, \quad \sigma(a_1 a_2 \cdots a_m \cdots) = a_2 a_3 \cdots a_{m+1} \cdots$$

• the shift map is a good model for many properties in the theory of dynamical systems: widely studied examples

• Markov's original aim was modeling natural languages
A. A. Markov (1913), *Ein Beispiel statistischer Forschung am Text "Eugen Onegin" zur Verbindung von Proben in Ketten*

Shannon Entropy

- Claude E. Shannon, Warren Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, 1949. (reprinted 1998)

Entropy measures the uncertainty associated to a prediction of the result of the experiment (equivalently the amount of information one gains from performing the experiment)

- Shannon entropy of a Bernoulli measure

$$S(\mu_P) = -\sum_{a\in\mathfrak{A}} p_a \log(p_a)$$

- Entropy of a Markov measure

$$S(\mu_{\pi,P}) = -\sum_{a,b\in\mathfrak{A}} \pi_a p_{ab} \log(p_{ab})$$

(Kolmogorov–Sinai entropy)

Relative entropy and cross-entropy

- Relative entropy (Kullback–Leibler divergence) $P = (p_a)$, $Q = (q_a)$

$$KL(P||Q) = \sum_{a \in \mathfrak{A}} p_a \log(\frac{p_a}{q_a})$$

- cross entropy of probabilities $P = (p_a)$ and $Q = (q_a)$

$$S(P, Q) = S(P) + KL(P||Q) = -\sum_{a \in \mathfrak{A}} p_a \log(q_a)$$

expected message-length per datum when a wrong distribution Q is assumed while data follow distribution P

Entropy and Cross-entropy

• for a message W (thought of as a random variable) with $\mathcal{V}(W)$ set of possible values

$$S(W) = -\sum_{w \in \mathcal{V}(W)} \mathbb{P}(w) \log \mathbb{P}(w)$$

$$= \text{Average}_{w \in \mathcal{V}(W)}(\#\text{Bits Required for}(w))$$

• $\mathbb{P}_M$ = probabilities estimated using a model

$$S(W, \mathbb{P}_M) = -\sum_{w \in \mathcal{V}(W)} \mathbb{P}(w) \log \mathbb{P}_M(w)$$

cross-entropy

• Cross-entropy gives a model evaluator

Entropy and Cross-entropy of languages

- asymptotic limit of per-word entropy as length grows

$$S(\mathcal{L},\mathbb{P}) = -\lim_{n\to\infty} \frac{1}{n} \sum_{w\in\mathcal{W}^n(\mathcal{L})} \mathbb{P}(w)\log\mathbb{P}(w)$$

for language $\mathcal{L} \subset \mathfrak{A}^\star$ with $\mathcal{W}^n(\mathcal{L}) = \mathcal{L} \cap \mathfrak{A}^n$

- cross-entropy same with respect to a model $\mathbb{P}_M$

$$S(\mathcal{L},\mathbb{P},\mathbb{P}_M) = -\lim_{n\to\infty} \frac{1}{n} \sum_{w\in\mathcal{W}^n(\mathcal{L})} \mathbb{P}(w)\log\mathbb{P}_M(w)$$

Other notions of Entropy of languages

• language structure functions $s_{\mathcal{L}}(m) = \#\mathcal{W}^m(\mathcal{L})$ (number of strings of length m in the language)

• Generating function for the language structure functions

$$G_C(t) = \sum_m s_{\mathcal{L}}(m) t^m$$

• $\rho =$ radius of convergence of the series $G_C(t)$

• Entropy: $\mathcal{S}(\mathcal{L}) = -\log \rho(G_C(t))$

• Example: for $\mathcal{L} = \mathfrak{A}^\star$ with $\#\mathfrak{A} = N$, and $\mathbb{P}$ uniform distribution both $S(\mathcal{L}, \mathbb{P}) = \mathcal{S}(\mathcal{L}) = \log N$

Trigram model

• can consider further dependencies between letters beyond consecutive ones

• Example trigram: how next letter in a word depends on the previous two

$$\mathbb{P}(w) = \mathbb{P}(w_0)\mathbb{P}(w_1|w_0)\mathbb{P}(w_2|w_0 w_1)\cdots\mathbb{P}(w_m|w_{m-2}w_{m-1})$$

• build the model probabilities by counting frequencies of sequences $w_{j-2}w_{j-1}w_j$ of specific choices of three words over corpora of texts... problem: the model suppresses grammatical but unlikely combinations of words, which occur infrequently
problem of sparse data

• possible solution: smoothing out the probabilities

$$\mathbb{P}(w_m|w_{m-1}w_{m-2}) = \lambda_1\mathbb{P}_f(w_m)+\lambda_2\mathbb{P}_f(w_m|w_{m-1})+\lambda_3\mathbb{P}_f(w_m|w_{m-2}w_{m-1})$$

respectively frequencies $\mathbb{P}_f$ of single word, pair, *and* triple

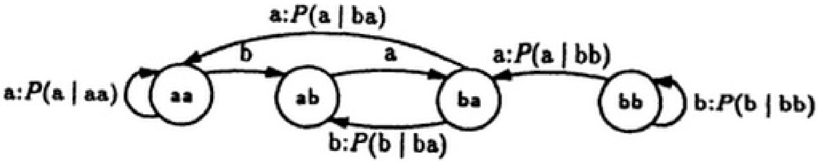
Hidden Markov Models

- these more general types of dependencies (like trigram) extend Markov Chains to Hidden Markov Models
- first step construct Markov Chain for

$$\mathbb{P}(w) = \prod_{j=0}^{m} \mathbb{P}(w_j | w_{j-2} w_{j-1})$$

by taking states consisting of pairs of consecutive letters $\mathfrak{A} \times \mathfrak{A}$ and an edge for each $a \in \mathfrak{A}$ with probability of transition $\mathbb{P}(c|ab)$ from ab to bc

Example:



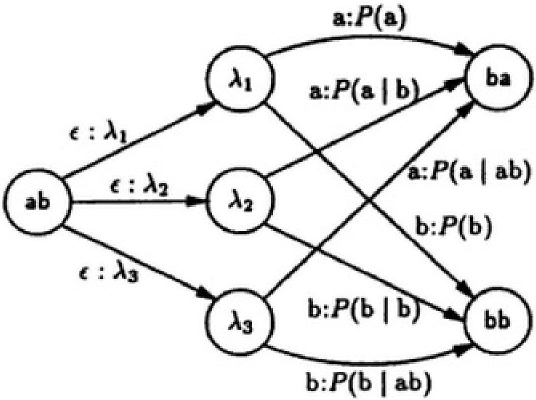
- second step: introduce the combination (with $\sum_i \lambda_i = 1$)

$$\lambda_1 \mathbb{P}(w_m) + \lambda_2 \mathbb{P}(w_m | w_{m-1}) + \lambda_3 \mathbb{P}(w_m | w_{m-2} w_{m-1})$$

in the diagram by replacing edges out of every node of the Markov chain with a diagram with additional states marked by λ_i and additional edges corresponding to all the probabilities contributing: $\mathbb{P}(a)$, $\mathbb{P}(a|b)$ and $\mathbb{P}(c|ab)$ with edges into λ_i state marked by empty ϵ output symbol ...

- the presence of ϵ-transitions with no output symbol implies this is a Hidden Markov Model with hidden states and visible states

Example: replacing the part of the diagram connecting node *ab* to nodes *ba* and *bb*



Shannon's series approximations to English

Random texts composed according to probability distributions that better approximate English texts

• Step 0: random text from English alphabet (plus blank symbol): letters drawn with uniform Bernoulli distribution

XFOML RXKHRJFFJUJ ZLPWCFWKCYJ
FFJEYVKCQSGXYD QPAAMKBZAACIBZLHJQD

• Step 1: random text with Bernoulli distribution based on frequency of letters in English

OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI
ALHENHTTPA OOBTTVA NAH BRL

- Step 2: random text with Markov distribution over two consecutive letters from English frequencies (diagram model)

ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

- Step 3: trigram model

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE

Can also do the same on an alphabet of words instead of letters

- Step 1: random text with Bernoulli distribution based on frequency of words in sample texts of English

> REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

- Step 2: Markov distribution for consecutive words (digram)

> THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED

Stone house poet Mr.Shih who ate ten lions

石 室 詩 士 施 氏 嗜 獅 誓 食 十 獅 氏 時 時 適 市 視 獅 十 時 氏
適 市 適 十 碩 獅 適 市 是 時 氏 視 是 十 獅 恃 十 石 矢 勢 使 是
十 獅 逝 世 氏 拾 是 十 獅 屍 適 石 室 石 室 濕 氏 使 侍 試 拭 石
室 石 室 拭 氏 始 試 食 是 十 獅 屍 食 時 始 識 是 十 碩 獅 屍 實
十 碩 石 獅 屍 是 時 氏 始 識 是 實 事 實 試 釋 是 事

Text presented at the tenth Macy Conference on Cybernetics (1953) by linguist Yuen Ren Chao

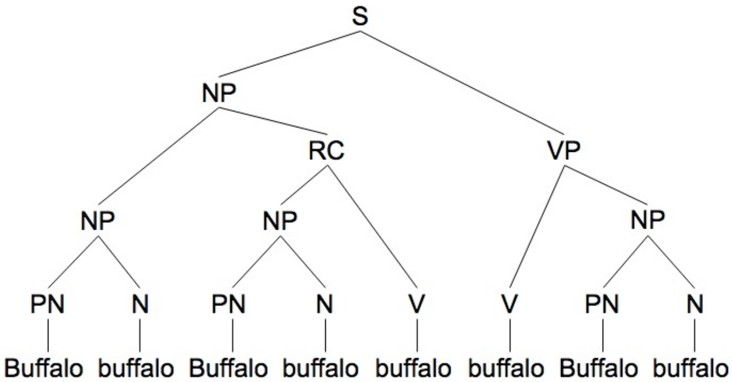
- Shannon information as a written text very different from Shannon information as a spoken text

- for an interesting example in English, consider the word *buffalo*

\- *buffalo!* = bully! VP
\- *Buffalo buffalo* = bison bully NP VP
\- *Buffalo buffalo buffalo* = those bison that are from the city of Buffalo bully
\- *Buffalo buffalo buffalo buffalo* = those bison that are from the city of Buffalo bully other bison

- in fact arbitrarily long sentences consisting solely of the word *buffalo* are grammatical

(Example by Thomas Tymoczko, logician and philosopher of mathematics)



Bison from Buffalo, that bison from Buffalo bully, themselves bully bison from Buffalo

Where is the information content? as string (in an alphabet of words) it has Shannon information zero... the information is in the parse trees

Probabilistic Context Free Grammars $\mathcal{G} = (V_N, V_T, P, S, \mathbb{P})$

- V_N and V_T disjoint finite sets: *non-terminal* and *terminal* symbols
- $S \in V_N$ *start symbol*
- P finite rewriting system on $V_N \cup V_T$

$P =$ *production rules* $A \to \alpha$ with $A \in V_N$ and $\alpha \in (V_N \cup V_T)^\star$

- Probabilities $\quad \mathbb{P}(A \to \alpha)$

$$\sum_\alpha \mathbb{P}(A \to \alpha) = 1$$

ways to expand same non-terminal A add up to probability one

Probabilities of parse trees

• $\mathcal{T}_{\mathcal{G}} = \{T\}$ family of parse trees T for a context-free grammar $\mathcal{G}$

• if $\mathcal{G}$ probabilistic, can assign probabilities to all the possible parse trees $T(w)$ for a given string w in $\mathcal{L}_{\mathcal{G}}$

$$\mathbb{P}(w) = \sum_{T=T(w)} \mathbb{P}(w,T) = \sum_{T} \mathbb{P}(T)\,\mathbb{P}(w|T) = \sum_{T=T(w)} \mathbb{P}(T)$$

last because tree includes the terminals (labels of leaves) so $\mathbb{P}(w|T(w)) = 1$

• Probabilities account for syntactic ambiguities of parse trees in context-free languages

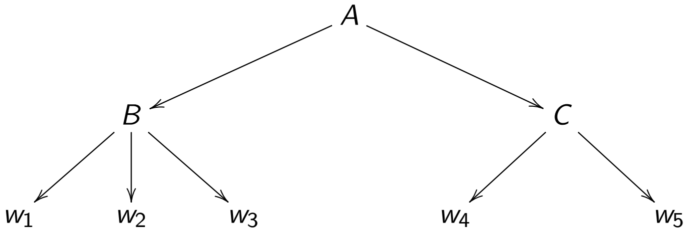
Subtree independence assumption

• a vertex v in an oriented rooted planar tree T *spans* a subset $\Omega(v)$ of the set of leaves of T if $\Omega(v)$ is the set of leaves reached by an oriented path in T starting at v

• denote by A_{kl} a non-terminal labeling a vertex in a parse tree T that spans the subset $w_k \dots w_l$ of the string $w = w_1 \dots w_n$ parsed by $T = T(w)$

1. $\mathbb{P}(A_{kl} \to w_k \dots w_l \mid \text{anything outside of } k \leq j \leq l\} = \mathbb{P}(A_{kl} \to w_k \dots w_l)$
2. $\mathbb{P}(A_{kl} \to w_k \dots w_l \mid \text{anything above } A_{kl} \text{ in the tree}\} = \mathbb{P}(A_{kl} \to w_k \dots w_l)$

Example



$$\mathbb{P}(T) = \mathbb{P}(A, B, C, w_1, w_2, w_3, w_4, w_4 \mid A)$$

$$= \mathbb{P}(B, C \mid A)\, \mathbb{P}(w_1, w_2, w_3 \mid A, B, C)\, \mathbb{P}(w_4 w_5 \mid A, B, C, w_1, w_2, w_3)$$

$$= \mathbb{P}(B, C \mid A)\, \mathbb{P}(w_1, w_2, w_3 \mid B)\, \mathbb{P}(w_4 w_5 \mid C)$$

$$= \mathbb{P}(A \to BC)\, \mathbb{P}(B \to w_1, w_2, w_3)\, \mathbb{P}(C \to w_4 w_5)$$

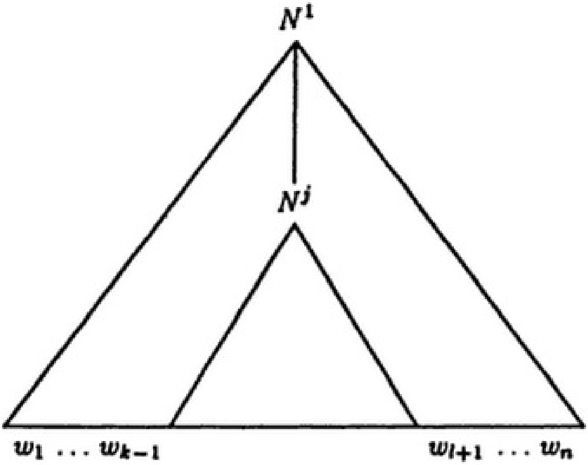
Sentence probabilities in PCFGs

• Fact: context-free grammars can always be put in Chomsky normal form where all the production rules are of the form

$$N \to w, \qquad N \to N_1 N_2$$

where N, N_1, N_2 are non-terminal, w terminal

• Parse trees for a CFG in Chomsky normal form have either an internal node marked with non-terminal N and one output to a leaf with terminal w or a node with nonterminal N and two outputs with non-terminals N_1 and N_2


N^1
N^j
$w_1 \dots w_{k-1}$
$w_{l+1} \dots w_n$

- assume CFG in Chomsky normal form
- inside probabilities

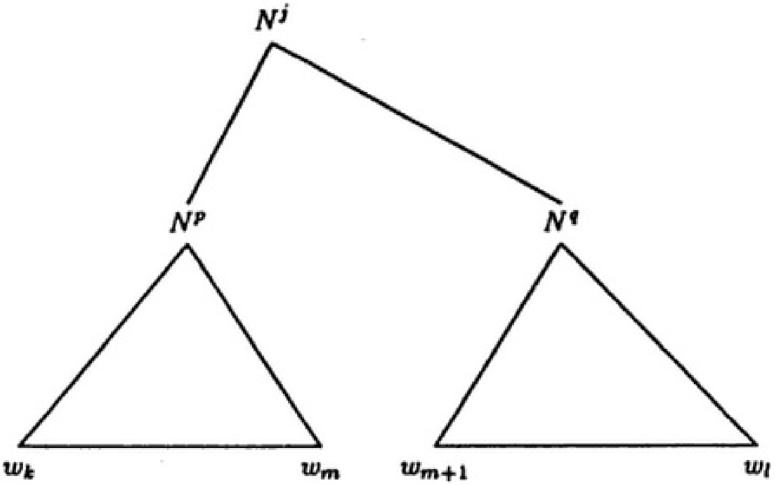
$$\beta_j(k,\ell) := \mathbb{P}(w_{k,\ell} \,|\, N^j_{k,\ell})$$

probability of the string of terminals “inside” (outputs of) the oriented tree with vertex (root) $N^j_{k,\ell}$

- outside probabilities

$$\alpha_j(k,\ell) := \mathbb{P}(w_{1,k-1}, N^j_{k,\ell}, w_{\ell+1,n})$$

probability of everything that's outside the tree with root $N^j_{k,\ell}$


N^j
N^p
N^q
w_k
w_m
w_{m+1}
w_l

Recursive formula for inside probabilities

$$\beta_j(k,\ell) = \mathbb{P}(w_{k,\ell} \mid N^j_{k,\ell}) = \sum_{p,q,m} \mathbb{P}(w_{k,m}, N^p_{k,m} w_{m+1,\ell} N^q_{m+1,\ell} \mid N^j_{k,\ell})$$

$$= \sum_{p,q,m} \mathbb{P}(N^p_{k,m}, N^q_{m+1,\ell} \mid N^j_{k,\ell}) \cdot \mathbb{P}(w_{k,m} \mid N^j_{k,\ell}, N^p_{k,m}, N^q_{m+1,\ell})$$

$$\cdot \mathbb{P}(w_{m+1,\ell} \mid w_{k,m}, N^j_{k,\ell} N^p_{k,m}, N^q_{m+1,\ell})$$

$$= \sum_{p,q,m} \mathbb{P}(N^p_{k,m}, N^q_{m+1,\ell} \mid N^j_{k,\ell}) \cdot \mathbb{P}(w_{k,m} \mid N^p_{k,m}) \cdot \mathbb{P}(w_{m+1,\ell} \mid N^q_{m+1,\ell})$$

$$= \sum_{p,q,m} \mathbb{P}(N^j \to N^p N^q) \cdot \beta_p(k,m) \beta_q(m+1,\ell)$$

Training Probabilistic Context-Free Grammars

• simpler case of a Markov chain: consider a transition $s^i \xrightarrow{w^k} s^j$ from state s^i to state s^j labeled by w^k

• given a large training corpus: count number of times the given transition occurs: counting function $C(s^i \xrightarrow{w^k} s^j)$

• model probabilities on the frequencies obtained from these counting functions:

$$\mathbb{P}_M(s^i \xrightarrow{w^k} s^j) = \frac{C(s^i \xrightarrow{w^k} s^j)}{\sum_{\ell,m} C(s^i \xrightarrow{w^m} s^\ell)}$$

• a similar procedure exists for Hidden Markov Models

• in the case of Probabilistic Context Free Grammars: use training corpus to estimate probabilities of production rules

$$\mathbb{P}_M(N^i \to w^j) = \frac{C(N^i \to w^j)}{\sum_k C(N^i \to w^k)}$$

• At the internal (hidden) nodes counting function related to probabilites by

$$C(N^j \to N^p N^q) := \sum_{k,\ell,m} \mathbb{P}(N^j_{k,\ell}, N^p_{k,m}, N^q_{m+1,\ell} \,|\, w_{1,n})$$

$$= \frac{1}{\mathbb{P}(w_{1,n})} \sum_{k,\ell,m} \mathbb{P}(N^j_{k,\ell}, N^p_{k,m}, N^q_{m+1,\ell}, w_{1,n})$$

$$= \frac{1}{\mathbb{P}(w_{1,n})} \sum_{k,\ell,m} \alpha_j(k,\ell) \mathbb{P}(N^j \to N^p N^q) \beta_p(k,m) \beta_q(m+1,\ell)$$

Training corpora and the syntactic-semantic interface

• the existence of different parse trees for the same sentence is a sign of semantic ambiguity

• training a Probabilistic Context Free Grammar over a large corpus can (sometime) resolve ambiguities by assigning different probabilities

• Example: two parsings of sentence: *They are flying planes* *They (are flying) planes* or *They are (flying planes)*. This type of ambiguity might not be resolved by training over a corpus

• Example: the three following sentences all have different parsings, but the most likely parsing can be picked up by training over a corpus

- *Time flies like an arrow*
- *Fruit flies like a banana*
- *Time flies like a stopwatch*

Probabilistic Tree Adjoining Grammars

- $\mathcal{I}$ = set of initial trees of the TAG; $\mathcal{A}$ = set of the auxiliary trees of the TAG
- each tree has subset of leaf nodes marked as nodes where substitution can occur
- adjunction can occur at any node marked by nonterminal (other than those marked for substitution)
- $s(\tau)$ set of substitution nodes of tree τ; $\alpha(\tau)$ set of adjunction nodes of τ
- $\mathcal{S}(\tau, \tau', \eta)$ = substitution of tree τ' into τ at node η;
$\mathcal{A}(\tau, \tau', \eta)$ = adjunction of tree τ' into tree τ at node η;
$\mathcal{A}(\tau, \emptyset, \eta)$ = no adjunction performed at node η
- Ω set of all substitutions and adjunction events

Probabilistic TAG (PTAG) $(\mathcal{I}, \mathcal{A}, \mathbb{P}_\mathcal{I}, \mathbb{P}_\mathcal{S}, \mathbb{P}_\mathcal{A})$

$$\mathbb{P}_\mathcal{I} : \mathcal{I} \to \mathbb{R}, \quad \mathbb{P}_\mathcal{S} : \Omega \to \mathbb{R}, \quad \mathbb{P}_\mathcal{A} : \Omega \to \mathbb{R}$$

$$\sum_{\tau \in \mathcal{I}} \mathbb{P}_\mathcal{I}(\tau) = 1$$

$$\sum_{\tau' \in \mathcal{I}} \mathbb{P}_\mathcal{S}(\mathcal{S}(\tau, \tau', \eta)) = 1, \quad \forall \eta \in s(\tau), \forall \tau \in \mathcal{I} \cup \mathcal{A}$$

$$\sum_{\tau' \in \mathcal{A} \cup \emptyset} \mathbb{P}_\mathcal{A}(\mathcal{A}(\tau, \tau', \eta)) = 1, \quad \forall \eta \in \alpha(\tau), \forall \tau \in \mathcal{I} \cup \mathcal{A}$$

- $\mathbb{P}_\mathcal{I}(\tau) =$ probability that a derivation begins with the tree τ
- $\mathbb{P}_\mathcal{S}(\mathcal{S}(\tau, \tau', \eta) =$ probability of substituting τ' into τ at η
- $\mathbb{P}_\mathcal{A}(\mathcal{A}(\tau, \tau', \eta)) =$ probability of adjoining τ' to τ at η

- Probability of a derivation in a PTAG:

$$\mathbb{P}(\tau) = \mathbb{P}_{\mathcal{I}}(\tau_0) \cdot \prod_{i=1}^{N} \mathbb{P}_{op_i}(op_i(\tau_{i-1}, \tau_i, \eta_i))$$

if τ obtained from initial tree τ_0 through a sequence of N substitutions and adjunctions op_i

- similar to probabilistic context-free grammars

Some References:

1. Eugene Charniak, *Statistical Language Learning*, MIT Press, 1996.
2. Rens Bod, Jennifer Hay, Stefanie Jannedy (Eds.), *Probabilistic Linguistics*, MIT Press, 2003.
3. Philip Resnik, *Probabilistic tree-adjoining grammar as a framework for statistical natural language processing*, Proc. of COLING-92 (1992) 418–424.