Partial Parsing

Steven Abney

University of Tübingen

abney@sfs.nphil.uni-tuebingen.de

A tutorial presen

Partial Parsing

- Standard parsers
  - Evaluate global parses, not partial parses
  - Do all-paths search (chart or no)
- Why unrestricted text is difficult
  - Incompleteness of lexicon
  - Incompleteness of grammar
  - Incompleteness of semantics
  - Long sentences
  - Errors in input
- Partial parsing
  - Produce forest
  - Speed
  - Reliability (precision)
  - Breadth
  - Robustness
  - Sacrifice depth of analys
- Levels
  - Breaking up "The Parsi
  - Fairly independent steps
  - Partial parsing is the
    tagging

Overview

Chunks

Cass
Chunks & dependencies
Supertags
Longest match
Finite-state
Chinks & chunks
Ejerhed
Church
Fidditch
Bourigault
Voutilainen
Chen & Chen
Rooth

MUC / IR

Futrelle
BBN
Seneff
AutoSlog
Fastus
Copsy

HMMs

Generation
Partial paths
NP recognition

Phrase Spotting

Relative likelihood
Alpha & beta

Parameter Estimation

Smoothing
Forward-backward

Finite-State grammars

HMMs are FSAs
Composing FSAs

Regre

Linear r
Regressi

Grammatic

Bayesian
Finch
Smith &

MI pa

Harris
Stolz
Magerman
Performanc

Lingu

Function
S-projec
Chunks &

Cass [6]

- Cascaded cheap analyzers
  1. Tag (Church tagger)
  2. First guess on NPs (Church NP-recognizer)
  3. Finite-state NP recognizer (correct some tagging and NP-boundary erro
  4. Chunks
  5. Simplex clauses
  6. Clause repair
  7. Attachment
- Each analyzer outputs a single 'best' answer
- Local search, but no global search, within levels
- Repair errors downstream

EOS	EOS
In$_{\mathrm{P}}$	$[_{\mathrm{PP}}$ In $[_{\mathrm{NP}}$ South Australia beds
[South$_{\mathrm{PN}}$ Australia$_{\mathrm{PN}}$ beds$_{\mathrm{NPl}}$]	
of$_{\mathrm{P}}$	$[_{\mathrm{PP}}$ of $[_{\mathrm{NP}}$ boulders]]
[boulders$_{\mathrm{NP}}$]	
were$_{\mathrm{Bed}}$	$[_{\mathrm{VP}}$ were deposited]
deposited$_{\mathrm{Vbn}}$	
by$_{\mathrm{P}}$	$[_{\mathrm{PP}}$ by $[_{\mathrm{NP}}$ melting icebergs]]
[melting$_{\mathrm{Vbg}}$ icebergs$_{\mathrm{NPl}}$]	
in$_{\mathrm{P}}$	$[_{\mathrm{PP}}$ in $[_{\mathrm{NP}}$ a gulf]]
[a$_{\mathrm{D}}$ gulf$_{\mathrm{N}}$]	
[that$_{\mathrm{Wps}}$]	$[_{\mathrm{WhNP}}$ that]
marked$_{\mathrm{Vbd}}$	$[_{\mathrm{VP}}$ marked]
[the$_{\mathrm{D}}$ position$_{\mathrm{N}}$]	$[_{\mathrm{NP}}$ the position]
of$_{\mathrm{P}}$	$[_{\mathrm{PP}}$ of $[_{\mathrm{NP}}$ the Adelaide geosyncl
[the$_{\mathrm{D}}$ Adelaide$_{\mathrm{PN}}$ geosyncline$_{\mathrm{N}}$]	
,	,
[an$_{\mathrm{D}}$ elongated$_{\mathrm{Vbn}}$]	$[_{\mathrm{NP}}$ an elongated, sediment-fille
,	
[sediment-filled$_{\mathrm{Vbn}}$ depression$_{\mathrm{N}}$]	
in$_{\mathrm{P}}$	$[_{\mathrm{PP}}$ in $[_{\mathrm{NP}}$ the crust]]
[the$_{\mathrm{D}}$ crust$_{\mathrm{N}}$]	
.	.
EOS	EOS

EOS
[$_{\text{PP}}$ In [$_{\text{NP}}$ South Australia beds]]
[$_{\text{PP}}$ of [$_{\text{NP}}$ boulders]]
[$_{\text{VP}}$ were deposited]
[$_{\text{PP}}$ by [$_{\text{NP}}$ melting icebergs]]
[$_{\text{PP}}$ in [$_{\text{NP}}$ a gulf]]

[$_{\text{WhNP}}$ that]
[$_{\text{VP}}$ marked]
[$_{\text{NP}}$ the position]
[$_{\text{PP}}$ of [$_{\text{NP}}$ the Adelaide geosyncline]]
,
[$_{\text{NP}}$ an elongated, sediment-filled depression]
[$_{\text{PP}}$ in [$_{\text{NP}}$ the crust]]

.
EOS

[NoSubj
EOS
[$_{\text{PP}}$ In [$_{\text{NP}}$ South Australi
[$_{\text{PP}}$ of [$_{\text{NP}}$ boulders]]
Pred: [$_{\text{VP}}$ were deposited]
[$_{\text{PP}}$ by [$_{\text{NP}}$ melting iceber
[$_{\text{PP}}$ in [$_{\text{NP}}$ a gulf]]
]
[SRC
Subj: [$_{\text{WhNP}}$ that]
Pred: [$_{\text{VP}}$ marked]
[$_{\text{NP}}$ the position]
[$_{\text{PP}}$ of [$_{\text{NP}}$ the Adelaide g
,
[$_{\text{NP}}$ an elongated, sedimen
[$_{\text{PP}}$ in [$_{\text{NP}}$ the crust]]
]
.
EOS

[NoSubj

EOS

[In South Australia]

S: [beds]

[of boulders]

P: [were deposited]

[by melting icebergs]

[in a gulf]

]

[SRC

S: [that]

P: [marked]

[the position]

[of the Adelaide geosyncline]

,

[an elongated, sediment-filled depression]

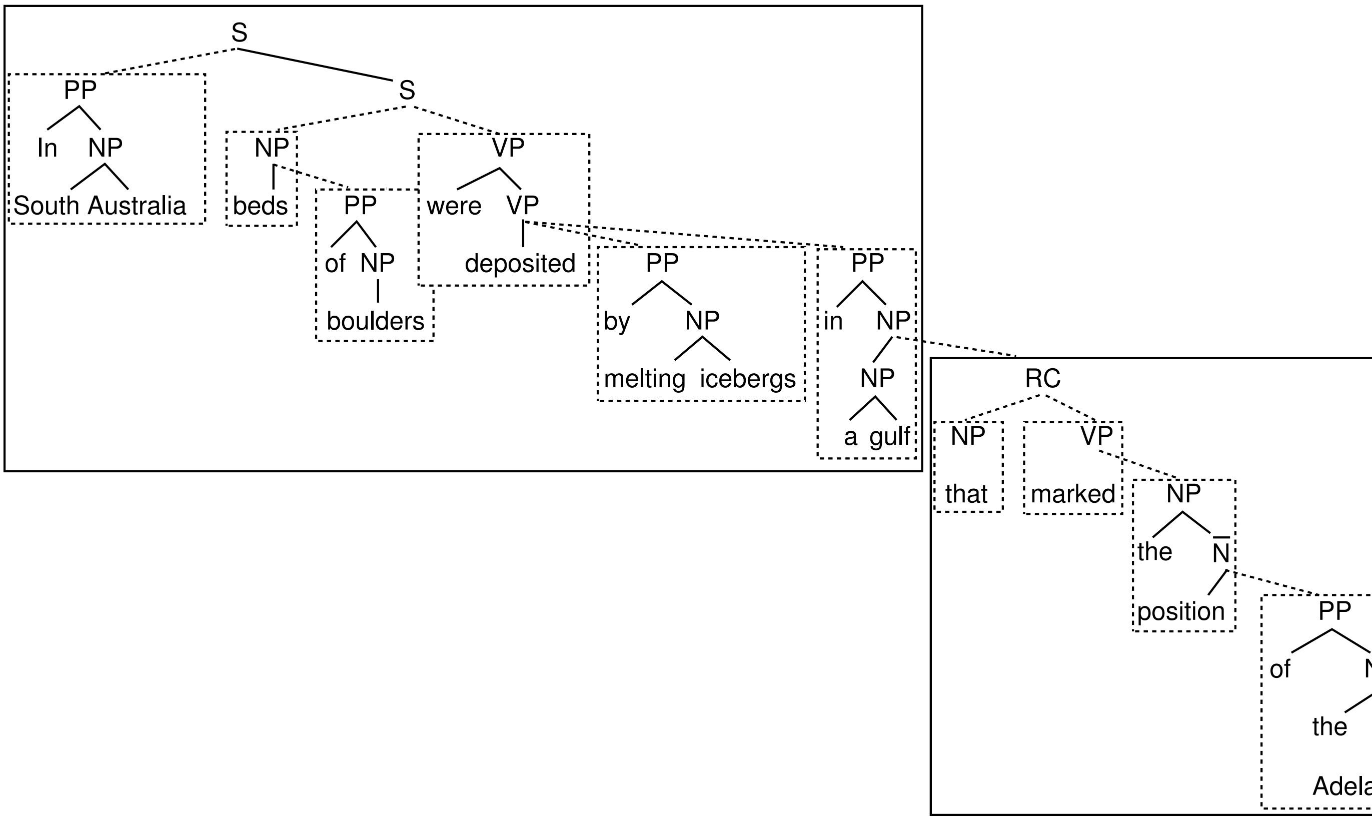
[in the crust]

]

.

EOS

Chunks and Dependencies

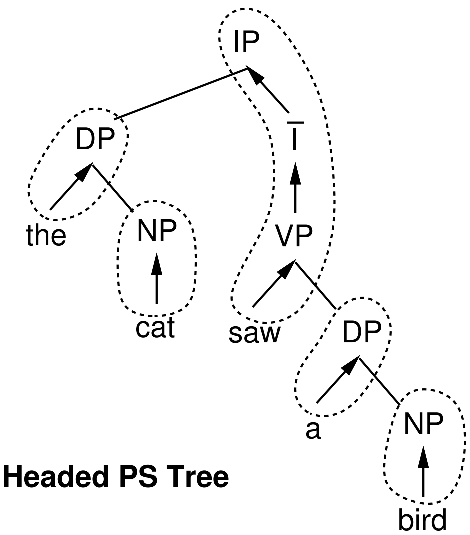


Chunks and Dependencies

- Factorization of the parsing problem
  - Dependencies: lexico-semantic, binary (head-dependent)
  - Chunks: syntactic category, finite-state sequences
- Simplex clauses
  - Trapping all-ways ambiguities
  - E.g., no PP-attachment across clause boundary
  - (Chunks trap noun-modification ambiguities)
- Instead of exponential global ambiguity, sequence of independent small sets

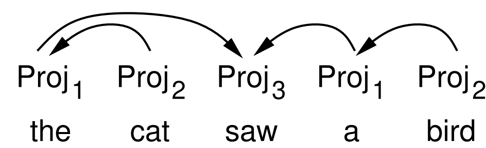
More generally

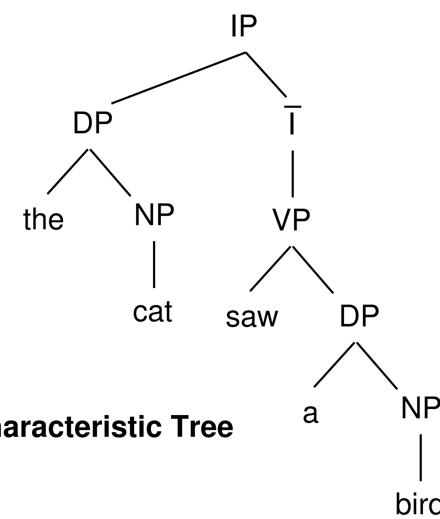
- Inspired by Gaifman [89]



Headed PS Tree

Projection-Dependency Tree

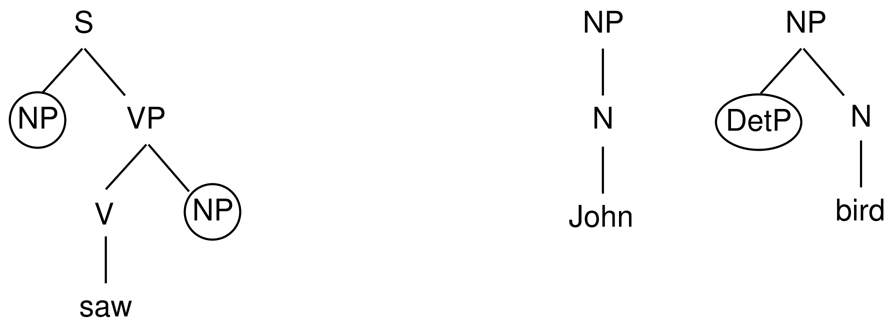




Characteristic Tree

Supertags

- Joshi & Srinivas [123]
- Instead of dependencies between projections, dependencies between elemen



- The difference: dependencies can also represent adjunction, not just substi
- Parsing as tagging: elementary trees are ‘supertags’
- Use standard tagging techniques (HMM’s)
- Or take advantage of dependency information in supertags to identify relevan 2-grams

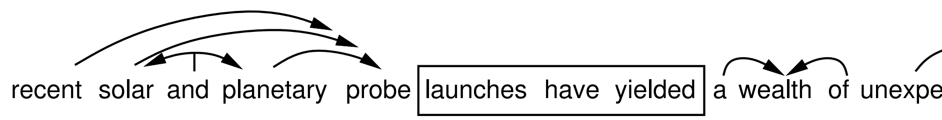
Voutilainen [191, 127]

- Variant of dependency grammar
- Parsing as tagging
  - Syntactic category tag
  - Syntactic function tag
- Rules are rules for eliminating tags ("constraints")

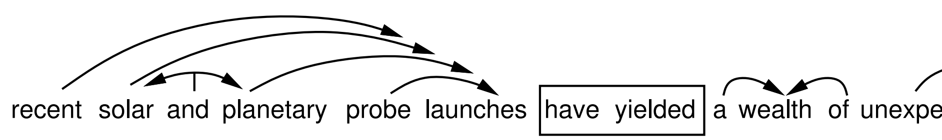
$$
\begin{array}{lcl}
\text{Vfin}\ldots & \rightarrow & \text{delete Ma} \\
\underline{\text{NomHead}}\ \&\ \ldots\text{Vfin}\ \&\ \neg\ \text{NomHead}\ldots\text{NomHead} & \rightarrow & \text{keeponly}
\end{array}
$$

- 1300 morphological rules, 120 syntactic rules
- Ambiguous representation

recent	>N
solar	>N
and	CC
planetary	>N
probe	NH
launches	V
have	V
yielded	V
a	>N
wealth	NH
of	<N
unexpected	>N
data	NH



recent	>N
solar	>N
and	CC
planetary	>N
probe	>N
launches	NH
have	V
yielded	V
a	>N
wealth	NH
of	<N
unexpected	>N
data	NH



Creative Ambiguity

- Or, Lazy Disambiguation
- Or, Picking the Fights You Can Win
- D-theory [150] — Say which clause a PP belongs ing where it's attached
- Unscoped quantificational formulae
- Ambiguity preservation in transfer in MT

Chunks

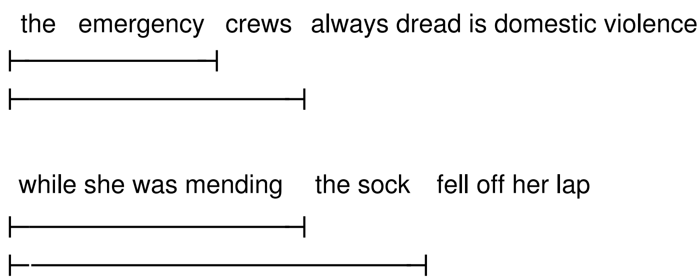
PP $\rightarrow$ (p | $\boxed{\text{to}}$)+ (NP | vbg)
WhPP $\rightarrow$ (p | to)+ WhNP
AdvP $\rightarrow$ (ql | precd | rb)* rb
AP $\rightarrow$ (AdvP | ql)* adj

Inf $\rightarrow$ to AdvP? VP-inf
VP $\rightarrow$ AdvP? (md | v-tns | hv-tns VPN? | be-tns (VPG | Vn)?)
VP-inf = AdvP? (vb | hv VPN? | be (VPG | Vn)?)
VPN = AdvP? (vbn | hvn | ben (VPG | Vn)?)
VPG = AdvP? (vbg | hvg | beg Vn?)
Vn = AdvP? (vbn | hvn | ben)

Other $\rightarrow$ any

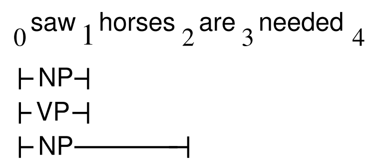
Longest match heuristic

- Used in lexical analyzers for compilers
- Psychologically plausible

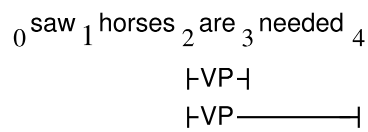


Longest Match

- One automaton for each phrase category
- Start automata at position i (initially, $i = 0$)
- Take longest match



- Set $i := j$ and repeat



Effectiveness of longest match

- Take chunks out of the UPenn Treebank

$$\begin{array}{lcl} \text{NP} & \rightarrow & \text{D N} \\ \text{NP} & \rightarrow & \text{D Adj N} \\ \text{VP} & \rightarrow & \text{V} \\ \text{VP} & \rightarrow & \text{Hv Vbn} \\ & \vdots & \end{array}$$

- At each point in string take longest matching pattern
  - Guess if multiple longest matches (of different category)
  - Punt one word if no match
- Performance:

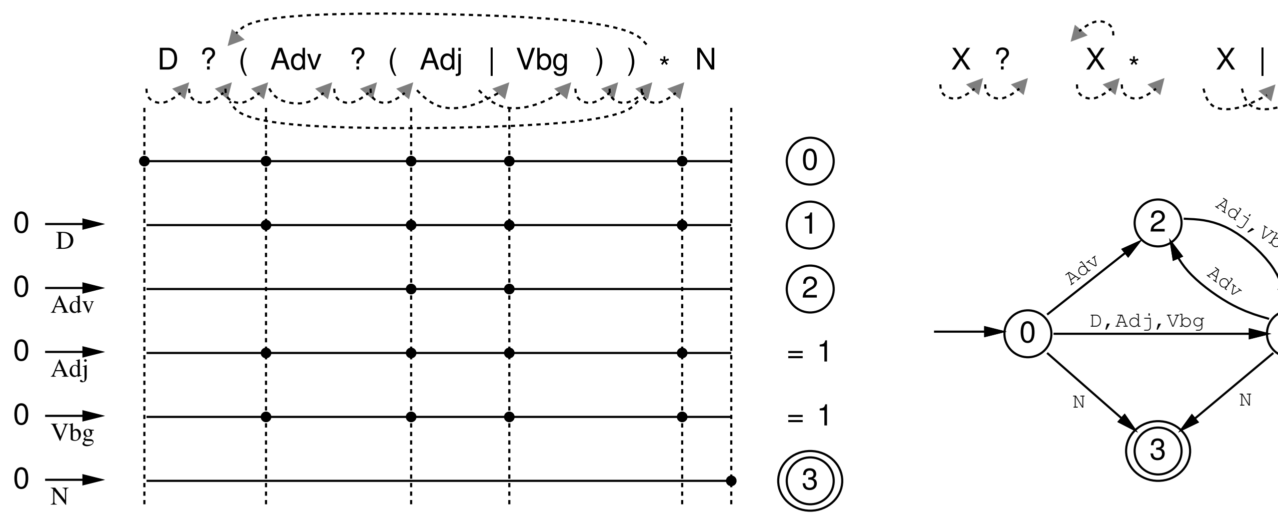
Precision	.92
Recall	.88

Finite-state techniques

- Hand-written grammar (regular expressions)

$$\text{NP} \rightarrow \text{Det? (Adj} \mid \text{Ing)} * \text{N}$$

- Compile into FSA



Clause

$$
\begin{array}{rcl}
\text{Extra-VPs} & \rightarrow & \text{EOC+ pre NP mid VP post (VP post)+} \\
\text{Clause} & \rightarrow & \text{EOC+ pre NP mid VP post} \\
\text{ObjRC} & \rightarrow & \text{EOC}{\ast}\ \text{WhNP pre NP mid VP post} \\
\text{SubjRC} & \rightarrow & \text{EOC}{\ast}\ \text{WhNP mid VP post} \\
\text{WhClause} & \rightarrow & \text{EOC}{\ast}\ \text{(WhPP} \mid \text{wrb) pre NP mid VP post} \\
\text{VP-Conj} & \rightarrow & \text{cc VP post} \\
\text{No-Subj} & \rightarrow & \text{EOC+ pre VP post} \\
\text{No-VP} & \rightarrow & \text{EOC+ post}
\end{array}
$$

$$
\begin{array}{rcl}
\text{pre} & = & \text{(X} \mid \text{Wh} \mid \text{PP-Conj)}{\ast}\ \text{((, AdvP)? ,)?} \\
\text{mid} & = & \text{(X} \mid \text{EOC-Soft} \mid \text{NP)}{\ast} \\
\text{post} & = & \text{(X} \mid \text{NP)}{\ast}
\end{array}
$$

$$
\begin{array}{rcl}
\text{PP-Conj} & = & \text{PP (, N PP}{\ast}\text{)}{\ast}\ \text{cc NP}
\end{array}
$$

$$
\begin{array}{rcl}
\text{X} & = & [\hat{}\ \text{Special}]
\end{array}
$$

$$
\begin{array}{rcl}
\text{Special} & = & \text{[EOC Wh NP VP]} \\
\text{EOC} & = & \text{[EOC-Hard EOC-Soft]} \\
\text{EOC-Hard} & = & \text{[: . eos]} \\
\text{EOC-Soft} & = & \text{[, cc cs that]} \\
\text{Wh} & = & \text{[WhNP WhPP wrb]}
\end{array}
$$

Bottom line

- Fast (once upon a time)

Pos:	4.2	ms/w
Cass:	15.0	ms/w
Total:	19.2	ms/w = 52 w/s

- Accurate

  ~5% error chunks
  ~5% error subj & pred

- BUT: Already in the tail
  - Only a few error types occur frequently
  - Only a few changes to the grammar will have much effect
  - The rest is sand

Parser speed

- Want a fast parser, get a fast machine
- Restricting search helps

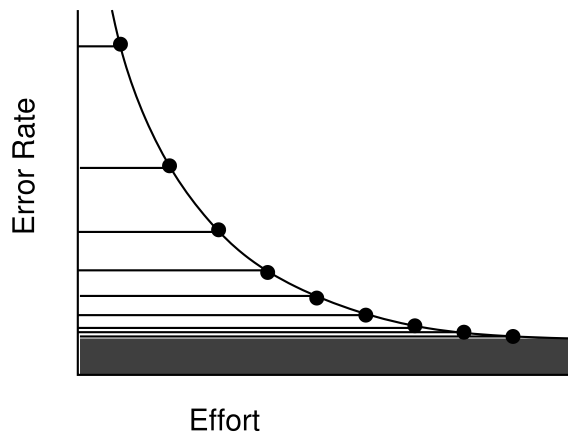
Program	depth	sw	hardware	w/s	
Fidditch3	parse	C	SGI	5600	
Copsy	np	Pascal	BS2000	2700	
CG	dep		Sparc10	1550	±250
Fidditch3	parse	C	Sun4	1200	
Pos	tag		Sun4	240	
Fidditch2	parse	Lisp	Sun4	62	
Cass	chunk	Lisp	Sun4	52	
Clarit	np	Lisp		50	
Fastus	chunk	Lisp	Sparc2	39	
Cass	chunk	Lisp	UX400S	32	
Scisor	skim			30	
Fidditch1	parse	Lisp	Sym-36xx	28	
McDonald	parse		MacII	14	±6
Chupa	parse	Lisp	UX400S	1.1	
Traditional	parse			0.20	

Lies, damned lies, and statistics

- What would you get by guessing?
  - Tagging: always taking most-frequent tag $\rightarrow$ 10% error
- Per-chunk error rate vs. per-sentence error rate

  5% chunk error
  10 chunks/sentence
  $1 - (1 - .5)^{10} = 40\%$ sentence error

- Zipf's Law
  - A little effort goes a long way—***at first***
  - The down side—further significant error reduction requires horrendous effort



Chinks and chunks

- Venerable idea:
  - Function words are phrase delimiters (chinks)
  - Content words are phrase contents (chunks)
- Ross & Tukey [164]
  - Used for sorting KWIC index of statistical works

on the $\boxed{\text{construction}}$ of $\boxed{\text{Bose-Chaudhuri matrices}}$
with the help of $\boxed{\text{Abelian group characters}}$

- fgroups
  - F+ C+
  - Used as low-level phrasal units in Bell Labs speech synthesizer

Ejerhed & Church [71, 70]

- Non-recursive (simplex) NP's and clauses
- Finite-state and stochastic methods
- Motivated in part by psycholinguistic studies
- Performance

	NP	Clause
Finite-state	3.3%	13%
Stochastic	1.4%	6.5%

- Application: text-to-speech (intonation)

Clause grammar

$$
\begin{array}{lcl}
\text{Clause} & \rightarrow & \text{cc? NP ([cc p \$] NP)}{*}\ \text{adv? tns-v X}{*}\ \text{Punct?} \\
 & | & \text{cc Adv? v X}{*}\ \text{Punct?} \\
 & | & \text{cc? Comp+ X}{*}\ \text{Punct?} \\
 & | & \text{cc? NP ([cc p \$] NP)}{*}\ \text{X}{*}\ \text{Punct?} \\
 & | & \text{Verb X}{*}\ \text{Punct?} \\
 & | & \text{cc? (Stray | NP)}{*}\ \text{X}{*}\ \text{Punct?} \\
 & & \\
\text{X} & = & [\hat{}\ \text{Comp Punct}] \\
\text{Comp} & = & [\text{cs to wdt wrb wps wpo wp\$ wql}] \\
\text{Punct} & = & [,\ .\ -\ :] \\
\text{Adv} & = & [\text{rb rbr}] \\
\text{Verb} & = & [\text{tns-v vbg vbn beg hvg}] \\
\text{Stray} & = & [\text{Adv rp ql neg nr jj jjr p}]
\end{array}
$$

Example

[the jury further said in term-end presentments]
[that the City Executive Committee ,]
[which had over-all charge of the election ,]
[deserves the praise and thanks of the City of Atlanta for the mann
[which the election was conducted .]

Church [57, 58]

- Stochastic tagger, followed by nonrecursive NP recognizer
- Between any pair of tags, we can insert one of:

`[` `]` `][` `-`

- Must keep track of whether inside or outside of NP

`[` the `[` corrosion weight loss `[`

- Computation:

B:	`[`	`-`	`-`	`-`	`]`	`[`	...
I:	0	1	1	1	1	0	...
T:	$D	DN	NN	NN	NP	PD	...

- Choose the sequence of brackets with the highest probability

Probabilities

B:	[	-	-	-	]	[	...
I:	0	1	1	1	1	0	...
T:	$D	DN	NN	NN	NP	PD	...

- Estimate by counting in parsed corpus $\hat{\Pr}(B|T) = \dfrac{f(B,T)}{f(T)}$

- Including inside/outside constraint

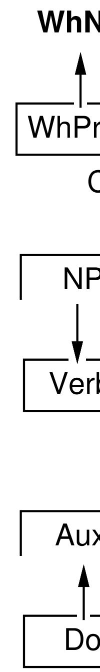
$$\begin{array}{ll} *[[& \Pr(B = b_{[}|T, I = 1) \\ *]] & \Pr(B = b_{]}|T, I = 0) \\ *]] & \Pr(B = b_{][}|T, I = 0) \end{array}$$

$$\Pr(B|T, I)$$

- Choices at different positions independent $\Pr(\mathbf{B}|\mathbf{T},\mathbf{I}) = \prod_i \Pr(\mathbf{B}_i|\mathbf{T}_i, \mathbf{I}_i$

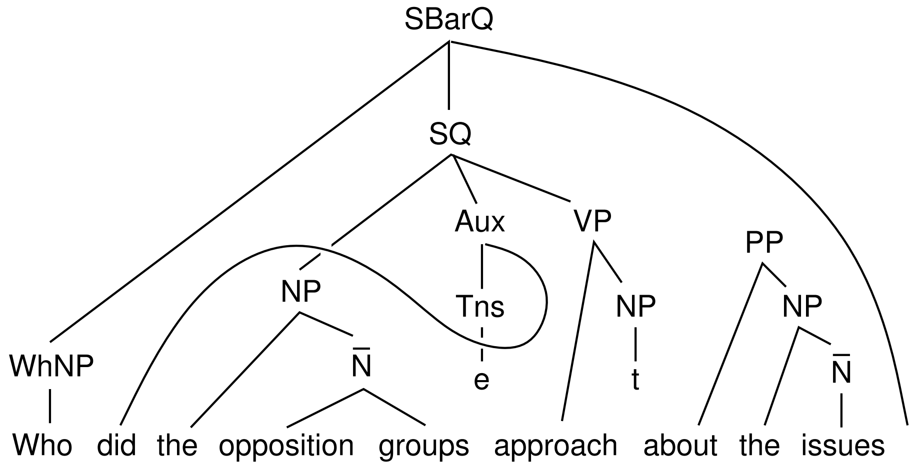
Fidditch [101]

- Industrial-strength version of Marcus Parser



Create	Recognizing leading edge of new node
Attach	Recognizing material belong to current node
Drop (Close)	Recognizing leading edge of material following node
Switch	Subject-aux inversion
Insert	Recognizing empty category
Attention-shift	Recognizing leading edge of NP in lookahead
Punt	Avoid an attachment decision

Fiddditch tree

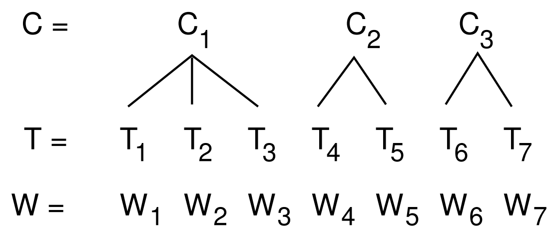


Bourigault [37]

- Extraction of likely multi-word terms for automatic indexing
- Phrase boundaries
  - Chinks: things that can't be chunks
  - E.g., Verbs, Pron, Conj, Prep (except *de, a*, Det
  - *un* [*traitement de texte*] *est installe sur le* [*disque dur de la stati*
- Parsing/extraction
  - Rules for extracting smaller potential terms
  - E.g. N_1 Adj P D N_2 P N_3 $\rightarrow$ N_1 Adj, N_2 P N_3
  - *disque dur, station de travail*
  - 800 such rules, manually built and tested

Chen and Chen [110]

- Building sequence of chunks on tags



- Best chunk

$$\begin{aligned} C* &= \mathrm{argmax}_C \Pr(C|W) \\ &\hat{=} \mathrm{argmax}_C \Pr(C|T) \\ &\hat{=} \mathrm{argmax}_C \prod_i \Pr(C_i|C_1, \ldots, C_{i-1}, T) \\ &\hat{=} \mathrm{argmax}_C \prod_i \Pr(C_i|C_{i-1}, T) \\ &\hat{=} \mathrm{argmax}_C \prod_i \Pr(C_i|C_{i-1}) \Pr(C_i|T) \qquad \textbf{!} \end{aligned}$$

- Probabilities estimated from parsed corpus (Susanne)

Internal probability and contextual probability

- Church and DeRose also say $\Pr(\mathbf{T}_i|\mathbf{T}_{i-1},\mathbf{W}_i) \hat{=} \Pr(\mathbf{T}_i|\mathbf{T}$
- Doesn't necessarily hurt performance
- But:

$$
\begin{aligned}
D &= \text{throw of die} \\
E &= 1 \text{ if } D \text{ is even, 0 otherwise} \\
L &= 1 \text{ if } D \leq 3 \text{, 0 otherwise}
\end{aligned}
$$

$\Pr(D=2|E=1,L=1)$

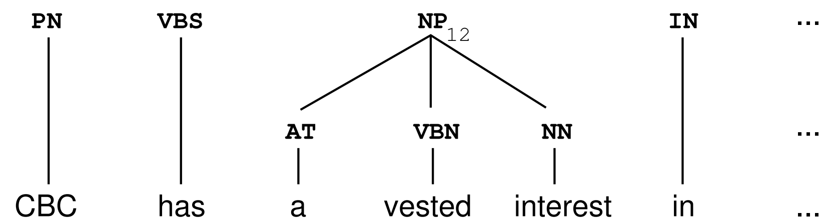
$\Pr(D=2|E=1)\Pr(D=2|L$

- Combining information sources: multivariate regression
- Alternative: HMM

$$
\begin{aligned}
\Pr(\mathbf{T}|\mathbf{W}) &\propto \Pr(\mathbf{T},\mathbf{W}) \\
&\hat{=} \prod_i \Pr(\mathbf{T}_i|\mathbf{T}_{i-1})\Pr(\mathbf{W}_i|\mathbf{T}_i)
\end{aligned}
$$

Rooth [163]

- Modified Hidden Markov Model



⋮ ⋮

$NP_{11} \rightarrow$ AT JJ NN

$NP_{12} \rightarrow$ AT VBN NN

$NP_{13} \rightarrow$ AT JJ NN NN

⋮ ⋮

- Generation probabilities $\Pr(x_i|x_{i-1})$ $\Pr(w|t)$
- Choose the structure by which the words were most likely generated

Threads

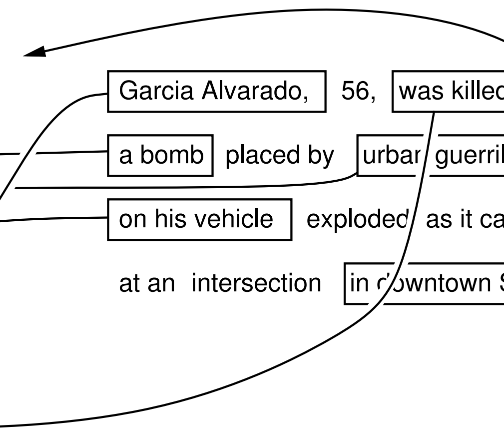
- Determinism
- Local evaluation of pieces
- Dependency grammar
  DG $\leftrightarrow$ CFG $\leftrightarrow$ chunks
- Levels/cascade
  - Specialized grammars
  - Creative ambiguity
- Longest match
- Likelihood
  - HMM's
  - Regression
- Induction (bootstrapping, C
- Linguistic/psycholinguistic

MUC

- Message Understanding Conference
- Task: data extraction from news reports
  - Filter out irrelevant texts
  - Tokenize and clean
  - Trigger on tokens
  - Fill semantic frames
  - Merge frames to fill data templates

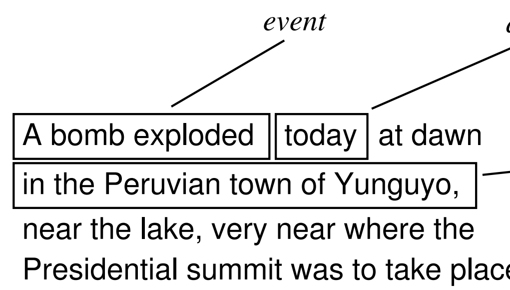
Example

0.	Message: ID	TST2-MUC4-0048
3.	Incident: Location	El Salvador: San Salvador (City)
4.	Incident: Type	Bombing
6.	Incident: Instrument ID	"bomb"
9.	Perp: Individual ID	"urban guerrillas"
12.	Phys Tgt: ID	"vehicle"
18.	Hum Tgt: Name	"Garcia Alvarado"
23.	Hum Tgt: Effect of Incident	Death: "Garcia Alvarado"



Weischedel et. al, “Partial Parsing” [195]

- Partial parsing for handling unrestricted text
- Message Understanding doesn’t require complete parse
  - Data extraction
  - Message routing
  - Message prioritization



- Questions
  - Effectiveness of fragment recognition?
  - How to interpret fragments?

- Interpretation
  - Identify headword to ge
    of phrase
  - Make attachment if cla
    requirement

Tokenize and clean

- Issues
  - Spelling errors
  - Foreign words / foreign names
  - Punctuation
  - Formulae
  - Graphics / Formatting
  - Sentence, paragraph boundaries
- Requirements
  - Fast
  - Highly reliable (snowball)
  - When in doubt, pass on ambiguity
- Shades into partial parsing

Futrelle *et al.* [87]

- Examples

7.3	sodium chloride
36,768	CO_2
2,6-diaminohexanoic acid	3.4×10^{-8}
^{3}H	

Cells were suspended in a medium containing $3.05 \times 10^{-2} \mu$M L-[*methyl*-^{3}H]-methione, α-methylaspartate and AIBU^8.

- Deterministic subgrammars
- Hand-correction

Examples

- Date/time expressions

24.10.94	10:06 p.m.
10/24/94	2000 GMT
Tues. the 24th Oct., 1994	two-thirty
Thu, 06 Oct 1994 11:47:55 EDT	

- Names
  - Person: *John T. Smith, Juan Mercedes Garcia de Mendoza, Kim*
  - Place: *the Orontes River; Mt. Pinatubo; Paris, TX*
  - Organization: *IBM; AT&T; Mt. Sinai Publishing Co., Inc.*
  - Titles: *Green County Sheriff's Deputy Gordon Caldwell*

- Bibliographic conventions

  Smyth (1990)
  Fig. 2
  ... as is probable.[6]
  NEW ORLEANS, 19 Jun 93 (API) –

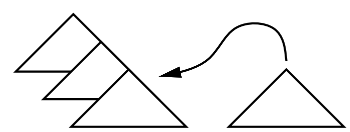
- State of the art: write little grammars by hand

PLUM (BBN) [17]

- Uses de Marcken parser to get fragments
- Semantic frames tied to words

$$\text{bomb}_V\,(\boldsymbol{subj}\,[1],\ \boldsymbol{obj}\,[2]) \begin{bmatrix} \text{bombing} & \\ \text{ti-perp-of} & \langle\text{person}\rangle\ [\![1]\!] \\ \text{object-of} & \langle\text{any}\rangle\ [\![2]\!] \end{bmatrix}$$

- Frame of fragment is gotten from head
- Assemble fragments deterministically via *attachment*
  - Try leftward attachments first
  - Try low attachments before high
  - Take first attachment satisfying slot constraints



Seneff [174]

- Start with standard full-sentence parser
- Parse fails: no $S[0, n]$
  - Consider $X[i, j]$ for X "major" and $i = 0$
  - Take longest match (maximize j)
  - Set $i = j$, repeat
  - If no $X[i, j]$, take next word, set $i = i + 1$, repeat
- Use discourse processor to integrate fragments
- Bottom line: good, but not as good as full-sentence parser

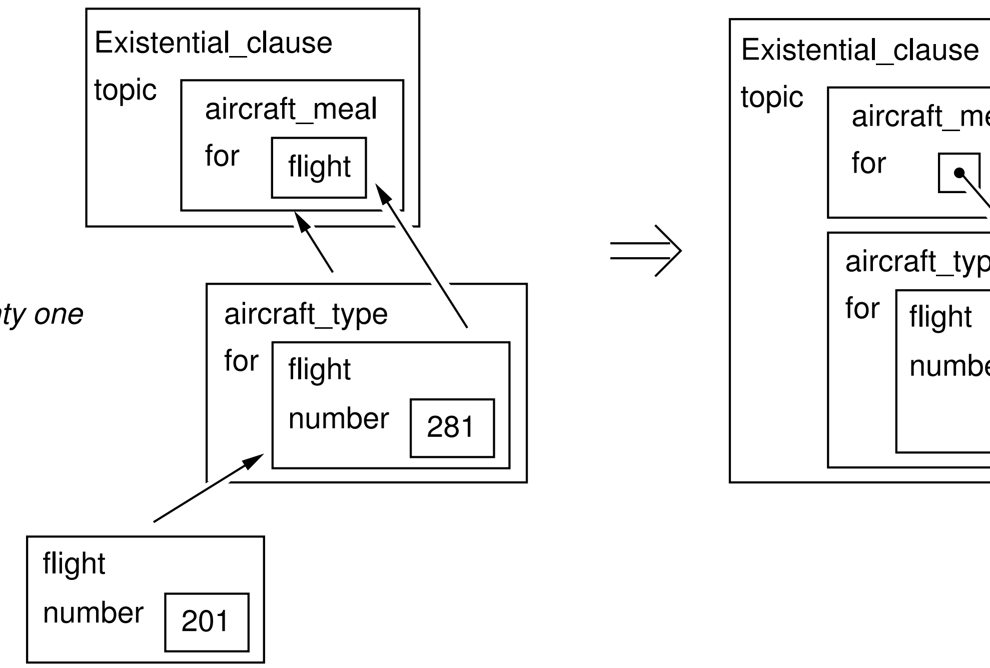
Using Discourse Processor

(what are the meals) and (aircraft for flight two eighty one) and also for (flight two oh one)

what are the meals

aircraft for flight two eighty one

flight two oh one



Big problem for frame-based systems

- Building lexicon of frames
- Frames provide robustness: assemble any way they fit
- Acquiring new frames from corpora
  - To name a few at random: [16, 34, 40, 44, 54, 60, 77, 95, 103, 128, 135
    176, 199]
- UMass: AutoSlog

AutoSlog [139]

- Input: examples of correct slot filles

  The ARCE battalion command has reported that about 50 peasants of various ages have been kidnapped by terrorists of the Farabundo Marti National Liberation Front in San Miguel department.

  [perp-indiv-id "terrorists"]

- Parse sentence, look at region around given word

actor:	peasants
verb:	kidnapped [PASSIVE]
prep:	by
pobj:	terrorists of FMNL

- Propose pattern

  $$\begin{array}{lcl} \text{verb} & = & \text{kidnapped [PAS} \\ \text{actor} & = & \text{ANY} \\ \text{PP}_{by} & = & \left[\begin{array}{l} \text{ORGANIZATI} \\ \text{TERRORIST} \\ \text{PROPER-NAM} \\ \text{HUMAN} \end{array}\right. \end{array}$$

- Automatic evaluation of precision/recall possible

Fastus [108, 107]

The inspiration for FASTUS was threefold. First, we were struck by the s
formance that the group at the University of Massachusetts got out of a fa
system. It was clear they were not doing anything like the depth of pre
syntactic analysis, or pragmatics that was being done by the systems at SR
Electric, or New York University. They were not doing a lot of process
were doing the *right* processing.

The second source of inspiration was Pereira's work on finite-state appro
of grammars, especially the speed of the implementation.

Speed was the third source. It was simply too embarassing to have to
the MUC-3 conference that it took TACITUS 36 hours to process 100
FASTUS has brought that time down to 11 minutes.

Fastus

- Triggering: single keywords from patterns plus known proper names
- Phrase recognition
  - Noun groups
  - Verb groups
  - P, Conj, RelPro, *ago*, *that*
  - Keep only longest match (nested, not overlapping)
- Patterns
  killing of $\langle$HumanTarget$\rangle$
  $\langle$GovtOfficial$\rangle$ accused $\langle$PerpOrg$\rangle$
  bomb was placed by $\langle$Perp$\rangle$ on $\langle$PhysicalTarget$\rangle$
- Merge compatible incidents

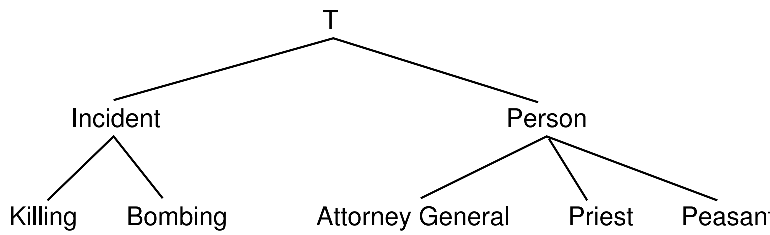
Fastus example

Noun Group:	Salvadoran President-elect
Name:	Alfredo Cristiani
Verb Group:	condemned
Noun Group:	the terrorist
Verb Group:	killing
Prep:	of
Noun Group:	Attorney General
Name:	Roberto Garcia Alvarado
Conj:	and
Verb Group:	accused
Noun Group:	the Farabundo Marti National Liberation Front (FM
Prep:	of
Noun Group:	the crime

Fastus merging

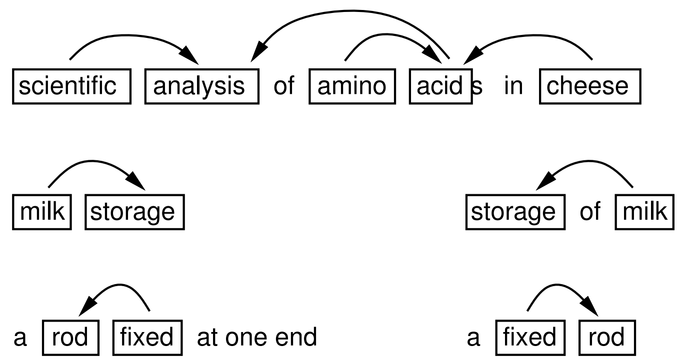
- Lots of frame scraps
- Merge if all slot-fillers compatible

$$\begin{bmatrix} \text{Incident} & \text{Killing} \\ \text{Perp} & – \\ \text{Confid} & – \\ \text{HumTarg} & \text{“Alvarado”} \end{bmatrix} + \begin{bmatrix} \text{Incident} & \text{Incident} \\ \text{Perp} & \text{FMLN} \\ \text{Confid} & \text{Suspected} \\ \text{HumTarg} & – \end{bmatrix} \Rightarrow \begin{bmatrix} \text{Incident} & \text{Killing} \\ \text{Perp} & \text{FMLN} \\ \text{Confid} & \text{Suspect} \\ \text{HumTarg} & \text{“Alvara} \end{bmatrix}$$



Schwarz: Copsy [169]

- Dependency parsing of noun phrases to improve precision in IR

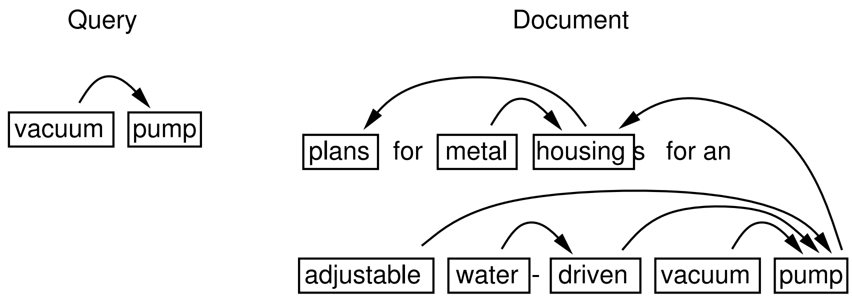


- Recognition rules must be
  - Relevant
  - Highly accurate
  - Cheap to apply

- Normalization
  - Dependencies
  - Development labor-inter
    200 proposed rules teste
    15,000 matching senten
    final rules

Copsy: matching

- Index only words, not phrases
- Presearch: boolean OR of words in query
- Parse query, match against parsed documents in initial return set



- Fast enough to parse documents at search time (19 Kb/s)
- Only 10% space overhead, however

More threads

- Interpretation
  - Dependencies $\leftrightarrow$ Slots
  - "class = head class" is consequence
  - Merging if slot-fillers are compatible
- Applications
  - Bootstrapping (collocations, alignment, ...)
  - MUC (Data extraction)
  - Terminology extraction
  - IR
  - Language models, spoken language understanding

Generation via Hidden Markov Model [160]

• Finite set of states	s_i
• Finite set of output symbols	w_i
• Random variables State at time t	Q_t
• Random variables Observation at time t	O_t
• Transition probabilities $\Pr(Q_{t+1} = s_j \mid Q_t = s_i)$	a_{ij}
• Emission probabilities $\Pr(O_t = w \mid Q_t = s_i)$	$b_i(w)$
• Initial probabilities $\Pr(Q_1 = s_i)$	π_i

Example: Tagger

- States are tags {\$, N, Pron, V, D}
- Output symbols are words {I, see, a, bird, .}
- Transition matrix

	\$	N	Pron	V	D
\$	0	.2	.5	0	.2
N	.3	.3	0	.4	0
Pron	.2	.1	0	.6	.1
V	.4	.2	.2	0	.2
D	0	1	0	0	0

- Emission matrix

	I	see	a	bird	.
\$	0	0	0	0	1
N	.1	.1	.1	.7	0
Pron	1	0	0	0	0
V	0	.9	0	.1	0
D	0	0	1	0	0

- Initial matrix

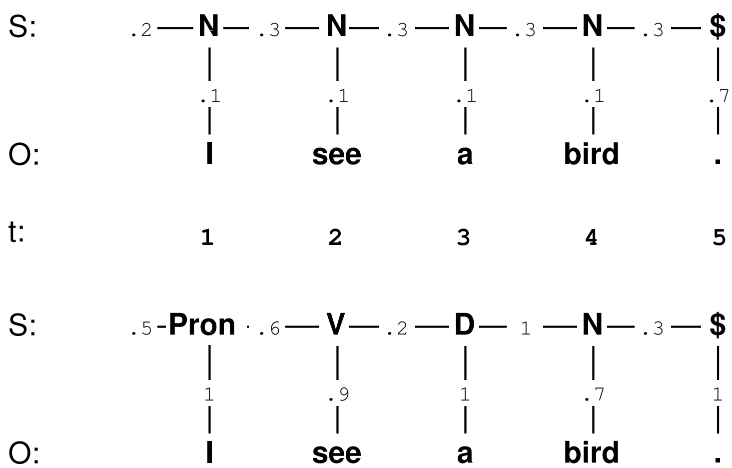
\$	N	Pron	V	D
0	.2	.5	0	.3

Probability of Generating a Structure

	$	N	Pron	V	D
$	0	.2	.5	0	.2
N	.3	.3	0	.4	0
Pron	.2	.1	0	.6	.1
V	.4	.2	.2	0	.2
D	0	1	0	0	0

	I	see	a	bird	.
$	0	0	0	0	1
N	.1	.1	.1	.7	0
Pron	1	0	0	0	0
V	0	.9	0	.1	0
D	0	0	1	0	0

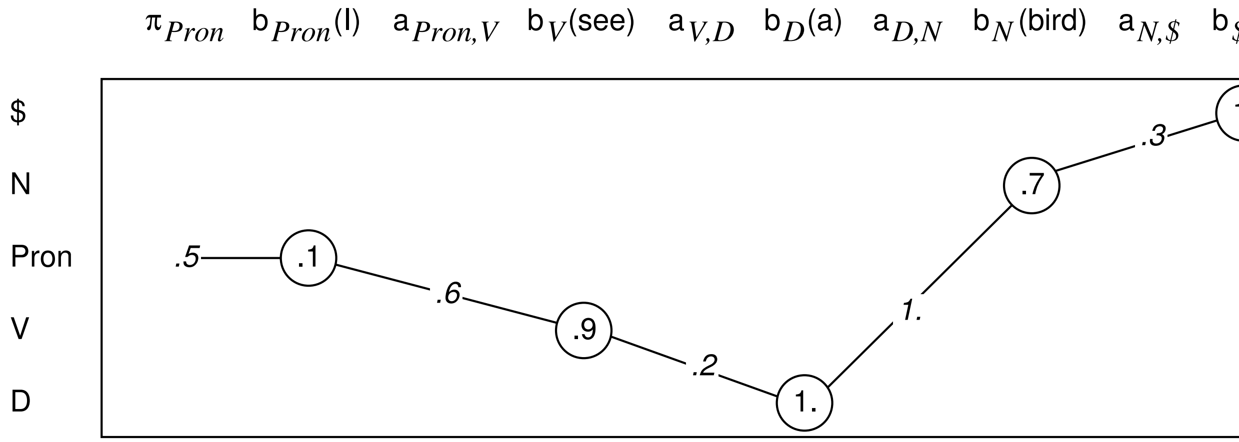
$
0



e^{-16}

$e^{-4.5}$

State sequence as path



Paths

State sequence (path)	$\mathbf{q}$	$=$	$(q_1, \ldots, q_T)$
Observation sequence	$\mathbf{o}$	$=$	$(o_1, \ldots, o_T)$
Probability	$\Pr(\mathbf{q}, \mathbf{o})$	$=$	$\Pr(Q_1 = q_1, \ldots, Q_T = q_T, O_1 = o_1, \ldots,$
Likelihood of path	$\mathrm{L}(\mathbf{q})$	$=$	$\Pr(\mathbf{q}, \mathbf{o})$

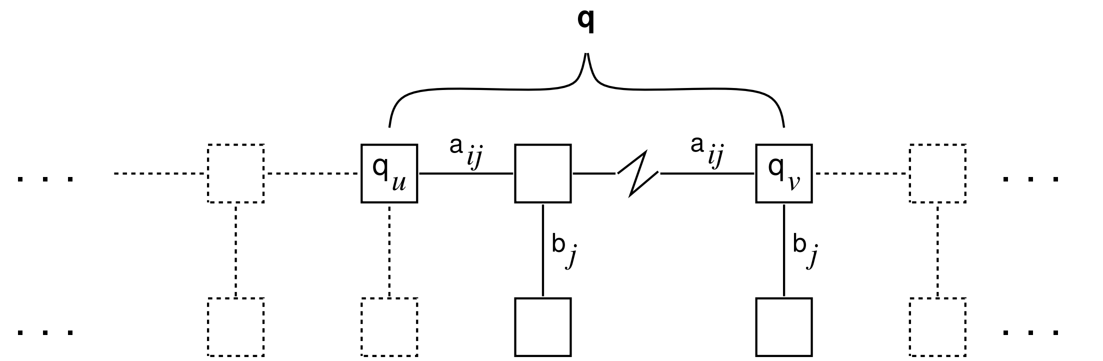
'Best' = Maximum Likelihood

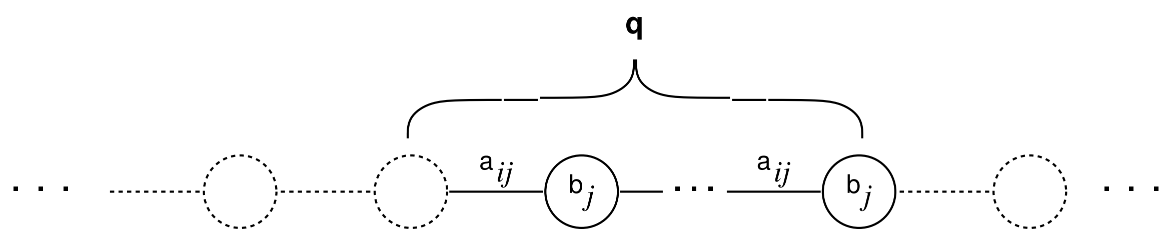
- We want $\mathbf{q}* = \operatorname*{argmax}_{\mathbf{q}} \Pr(\mathbf{q}|\mathbf{o})$
- By definition $\Pr(\mathbf{q}|\mathbf{o}) = \dfrac{\Pr(\mathbf{q},\mathbf{o})}{\Pr(\mathbf{o})}$
- Since $\Pr(\mathbf{o})$ is constant $\Pr(\mathbf{q}|\mathbf{o}) \propto \Pr(\mathbf{q},\mathbf{o})$
- Therefore $\operatorname*{argmax}_{\mathbf{q}} \Pr(\mathbf{q}|\mathbf{o}) = \operatorname*{argmax}_{\mathbf{q}} \Pr($
- Substituting $\mathbf{q}* = \operatorname*{argmax}_{\mathbf{q}} \mathrm{L}(\mathbf{q})$
- That is, $\mathbf{q}*$ is the maximum-likelihood state sequence

Partial paths

Partial path from u to v $\quad\quad \mathbf{q} : u, v$

Context-independent likelihood $\quad \lambda(\mathbf{q}) = \prod_{t=u}^{v-1} a_{ij} b_j(\mathbf{o}_{t+1})$ where $q_t = s_i, q$



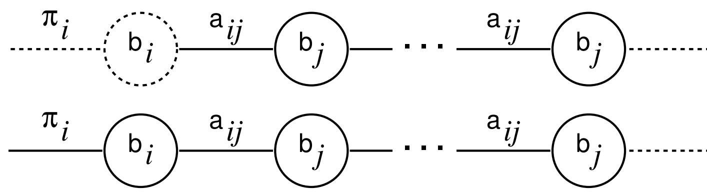


Partial paths

- Special case: initial

$$\mathbf{q} : 1, t \quad \lambda(\mathbf{q}) :$$

$$\lambda^0(\mathbf{q}) = \pi_i b_i(\mathbf{o}_1)\lambda(\mathbf{q})$$

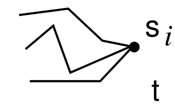


- Relation to likelihood

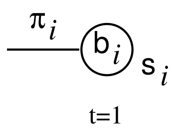
  if $\mathbf{q} : 1, T$ then $\mathrm{L}(\mathbf{q}) = \lambda^0(\mathbf{q})$

Computing $\mathbf{q}*$

- Most-likely partial sequence $\quad q_t^*(i) = \operatorname*{argmax}_{\mathbf{q}:1,t|\mathbf{q}_t=s_i} \lambda^0(\mathbf{q})$
- Likelihood thereof $\quad \delta_t(i) = \max_{\mathbf{q}:1,t|\mathbf{q}_t=s_i} \lambda^0(\mathbf{q})$

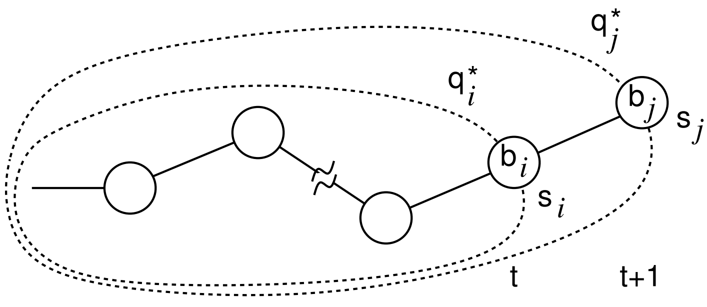


- Time 1



$$
\begin{aligned}
\delta_1(i) &= \pi_i b_i(\mathbf{o}_1) \\
q_1^*(i) &= \langle s_i \rangle
\end{aligned}
$$

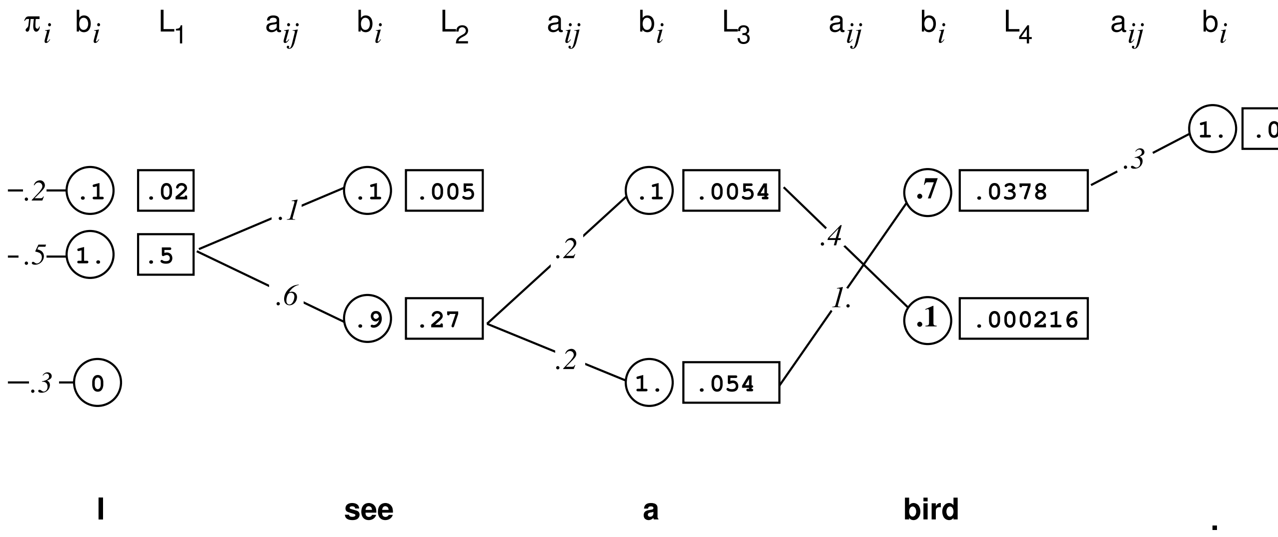
- Time $t+1$



$$
\begin{aligned}
\delta_{t+1}(j) &= \max_i \delta_t(i) a_{ij} b_j(\mathbf{o}_{t+} \\
i* &= \operatorname*{argmax}_i \delta_t(i) a_{ij} b_j(\\
q_{t+1}^*(j) &= q_t^*(i*)\,\hat{}\,\langle s_j \rangle
\end{aligned}
$$

Computing $\mathbf{q}*$

- Recursive definitions for $q_t^*(i)$, $\delta_t(i)$
- Fill in array by increasing values of variable of recursion (t)



NP-Recognizer as HMM

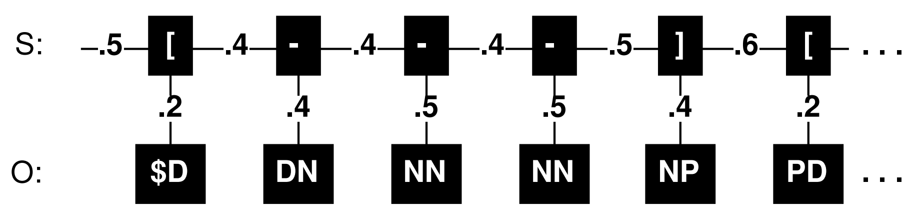
- States `[` `]` `I` `-` `#`
- π, a, b

[	]	I	-	#
.5				.5

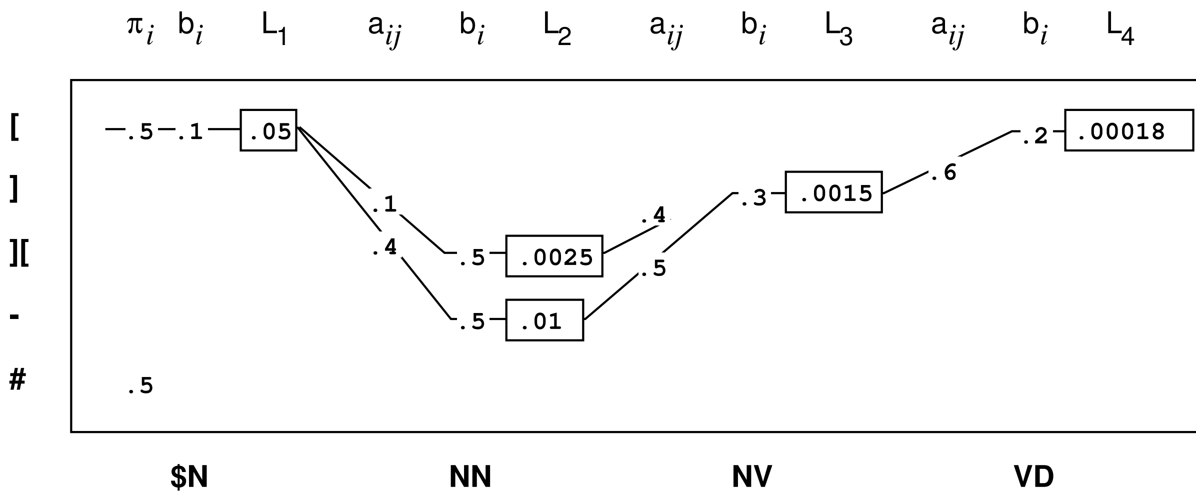
	[	]	I	-	#
[		.5	.1	.4	
]	.6				.4
I		.4		.6	
-		.5	.1	.4	
#	.2				.8

	$N	$D	$P	N$	NN	ND	NV	NP	DN	V$	VN	VD	VV	VP	P$	PN	PD	PV
[	.1	.2									.15	.2				.15	.2	
]				.25			.3	.4										.05
I					.5	.5												
-					.5			.1	.4									
#		.05	.3							.1			.1	.4	.05			

- $L(S)$



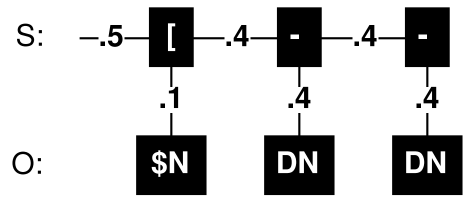
Viterbi with brackets



$$\left[\begin{array}{cc} \mathrm{N} & \mathrm{N} \\ \text{computer} & \text{science} \end{array} \right] \begin{array}{c} \mathrm{V} \\ \text{is} \end{array} \left[\begin{array}{cc} \mathrm{D} & \\ \text{a} & \ldots \end{array} \right.$$

Matching up pairs

- HMM does not guarantee that tag-pairs match up



- Define $\mathrm{L}'(\mathbf{q},\mathbf{o}) = \begin{cases} \alpha\mathrm{L}(\mathbf{q},\mathbf{o}) & \text{if } \mathbf{o} \text{ has matching tag-pairs} \\ 0 & \text{otherwise} \end{cases}$

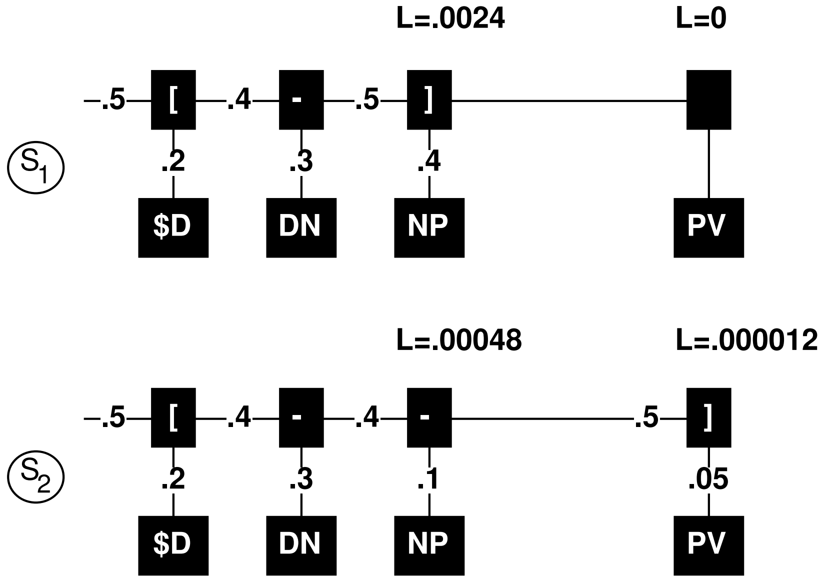
  – α is normalization constant to guarantee that

$$\sum_{\mathbf{q},\mathbf{o}} \mathrm{L}'(\mathbf{q},\mathbf{o}) = 1$$

Phrase spotting

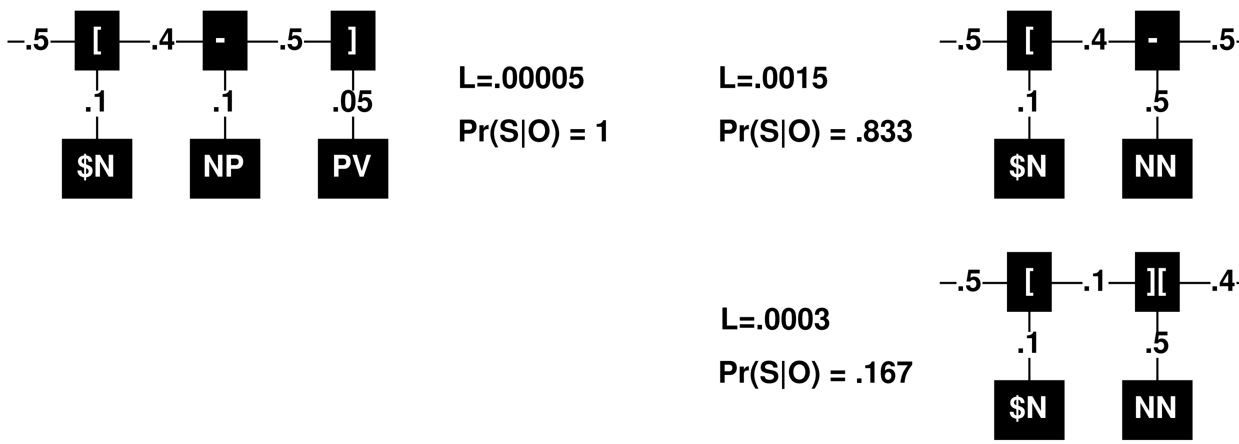
- Identifying individual phrases reliably
- E.g. for terminology extraction
- Aim: high precision, high recall, on individual phrases
  Don't care about getting complete, consistent parse for sentences
- Issues
  - Can't ignore context of candidate phrase
  - Can't directly compare $\lambda(\mathbf{q})$ and $\lambda(\mathbf{q'})$
  - How do we compute $\Pr(\mathbf{q}|\mathbf{o})$ for partial paths?

1. Can't just ignore context



Phrase spotting

2. Can't just compare likelihoods



Relative likelihood

- The likelihood of being right, given the input

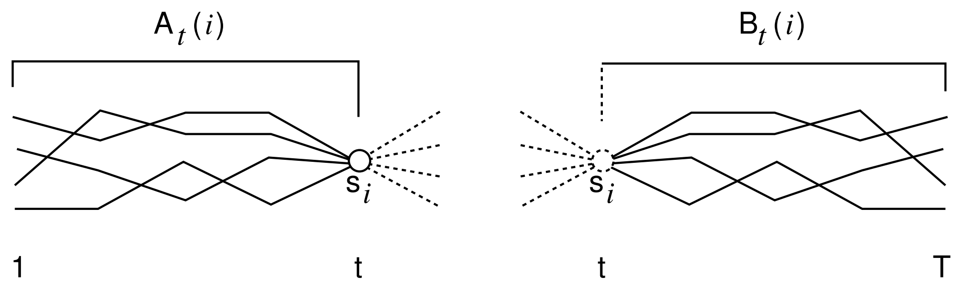
$$\begin{aligned} \Pr(\mathbf{q}|\mathbf{o}) &= \frac{\Pr(\mathbf{q},\mathbf{o})}{\Pr(\mathbf{o})} \\ &= \frac{\Pr(\mathbf{q},\mathbf{o})}{\sum_{\mathbf{q}'} \Pr(\mathbf{q}',\mathbf{o})} \\ &= \frac{\mathrm{L}(\mathbf{q})}{\sum_{\mathbf{q}'} \mathrm{L}(\mathbf{q}')} \end{aligned}$$

- For complete state-sequences, most-likely path is most-reliable

$$\underset{\mathbf{q}}{\operatorname{argmax}} \Pr(\mathbf{q},\mathbf{o}) = \underset{\mathbf{q}}{\operatorname{argmax}} \Pr(\mathbf{q}|\mathbf{o})$$

- Not so for partial paths

- Prefix and suffix paths

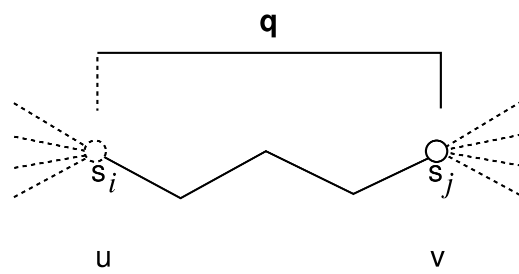


$$
\begin{aligned}
A_t(i) &= \{\mathbf{q} : 1, t | \mathbf{q}_t = s_i\} \\
\alpha_t(i) &= \sum_{\mathbf{q} \in A_t(i)} \lambda^0(\mathbf{q})
\end{aligned}
\qquad
\begin{aligned}
B_t(i) &= \{\mathbf{q} : t, T | \mathbf{q}_t = s \\
\beta_t(i) &= \sum_{\mathbf{q} \in B_t(i)} \lambda(\mathbf{q})
\end{aligned}
$$

Partial paths

- Partial-path likelihood

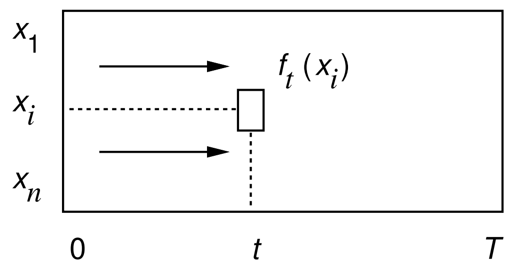
$$L(\mathbf{q}) = \Pr(\mathbf{q}, \mathbf{o}) = \alpha_u(i)\lambda(\mathbf{q})\beta_v(j)$$



- Relative likelihood

$$\Pr(\mathbf{q}|\mathbf{o}) = \frac{\mathrm{L}(\mathbf{q})}{\sum_{\mathbf{q\prime}:u,v} \mathrm{L}(\mathbf{q\prime})}$$

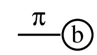
- $f_t(x_i)$ only requires values for $f_u(x_j)$ for $u < t$
- t is variable of recursion
- Fill in array by increasing t



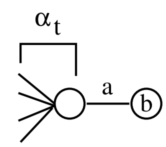
- Example: $\delta_t(i)$

Recursive definitions for α, β

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1)$$

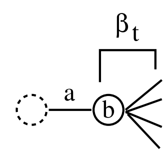


$$\alpha_{t+1}(j) = \sum_i \alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1})$$



$$\beta_T(i) = 1$$

$$\beta_{t-1}(i) = \sum_j a_{ij} b_j(\mathbf{o}_t) \beta_t(j)$$



Remaining problems for phrase-spotting

- Dependent on global analysis
  - Search is linear-time, but can be moderately expensive if large numbers
  - Poor enough models of 'garbage' can damage estimates of $\Pr(\mathbf{q}|\mathbf{o})$ for r
    $\mathbf{q}$
  - Can't always reliably segment text into sentences
- Integrating multiple information sources

Another problem: relative likelihood is not precision

- Some misspellings are undetectable at word level

  combing appositive NPs
  we had a rather milk winter

- Don't want to assume all words are misspelled (search)
- Would like to detect problem by low relative likelihood
- But if there's only one analysis, relative likelihood = 1, no matter how analysis
- Precision is corpus-global measure of relative likelihood

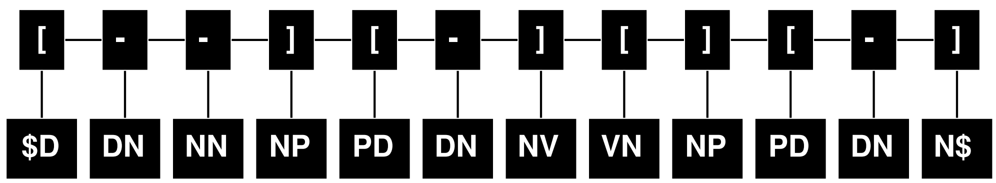
  E.g., of all the times we've seen "D Adv N N $", how often has it been an
- Have to estimate precision directly: it is neither likelihood nor relative likel

Parameter estimation

- With parsed corpus: count

$$
\begin{aligned}
a_{ij} &= \Pr(Q_{t+1} = s_j | Q_t = s_i) \;\hat{=}\; \frac{f(Q_t = s_i, Q_{t+1} = s_j)}{f(Q_t = s_i)} \\
b_i(w) &= \Pr(\mathbf{o}_t = w | Q_t = s_i) \;\hat{=}\; \frac{f(Q_t = s_i, \mathbf{o}_t = w)}{f(Q_t = s_i)}
\end{aligned}
$$

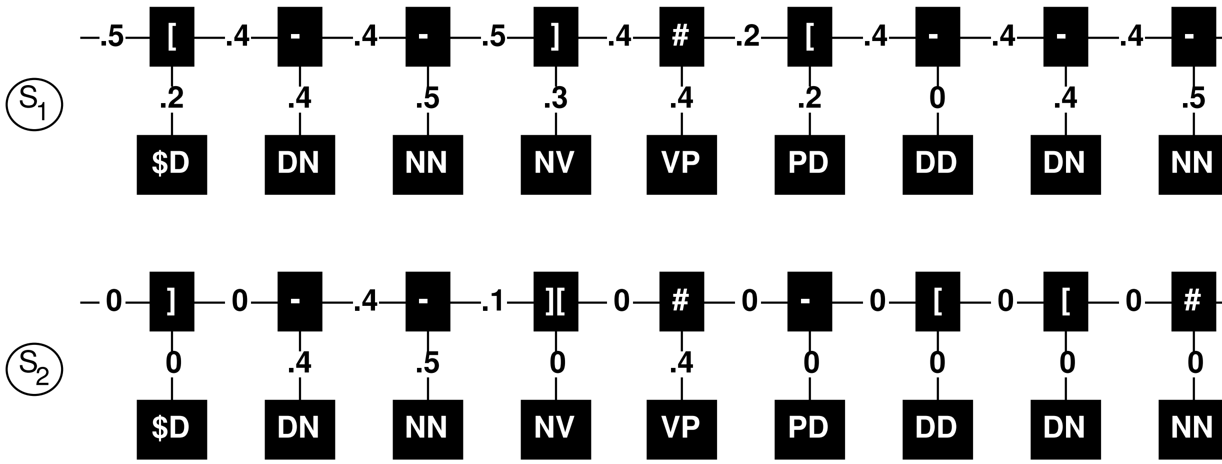
- Corpus is one giant observation sequence



$$\hat{a}_{[-} = \frac{f([-)}{f([)} = \frac{3}{4} \qquad \hat{b}_{[}(\mathrm{PD}) = \frac{f([, PD)}{f([)} = \frac{2}{4}$$

Why zeros are a problem

- Two structures with same likelihood: $L = 0$

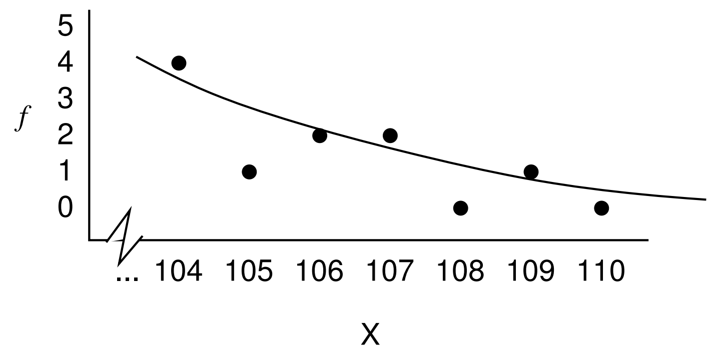


- But if we replace '0' with '.01':

$$L(S_1) = e^{-24}$$
$$L(S_2) = e^{-70}$$

Smoothing

- Choosing a good value to replace the zeros
- From choosing a smooth curve:



Good-Turing [59]

f	n_f	$f \cdot n_f$
9	$22,280$	$200,520$
8	$27,710$	$221,680$
7	$35,709$	$249,963$
6	$48,190$	$289,140$
5	$68,379$	$341,895$
4	$105,668$	$422,672$
3	$188,933$	$566,799$
2	$449,721$	$899,442$
1	$2,018,046$	$2,018,046$
0	$74,671,100,000$	0

$$\bar{f} \cdot n_f = (f+1) \cdot n_{f+1}$$

$$\bar{f} = \frac{(f+1) \cdot n_{f+1}}{n_f}$$

Cat-cal

- Categorize and calibrate
- Some of the events with 0 counts in training have > 0 counts in test
- Group by count

$$G_e = \{e' | f(e') = f(e)\}$$

- Re-estimate counts for groups from cross-validation corpus
- Re-estimate individual counts as group count times probability of choosing
  of group

$$\bar{f}(e) = \bar{f}(G_e) \cdot \Pr(e | G_e)$$

Cat-cal

		Corpus 1	Corpus2	$\bar{f}(G_i)$	$\Pr(e\|G_e)$	$\bar{}(f)$
G_2	[-	2	3	3	1	3
	[ ]	1	2		.3	1.2
G_1	- -	1	2	4	.3	1.2
	-]	1	0		.3	1.2
	][-	0	0		.2	.4
	# #	0	0		.2	.4
G_0	] #	0	0	2	.2	.4
	#]	0	0		.2	.4
	][ ]	0	0		.2	.4

Without Parsed Corpus

- Probability of transition from s_i to s_j at t to $t+1$

  $\Pr(Q_t = s_i, q_{t+1} = s_j | \mathbf{o}) = \Pr(\mathbf{q}|\mathbf{o})$ for $\mathbf{q} : t, t+1$, $\mathbf{q}_t = s_i$, $\mathbf{q}_{t+1} = s_j$

- Probability of being in s_i at t

  $\Pr(Q_t = s_i | \mathbf{o}) = \Pr(\mathbf{q}|\mathbf{o})$ for $\mathbf{q} : t, t$, $\mathbf{q}_t = s_i$

Without parsed corpus

- Use relative likelihood of transitions/emissions
- Suppose $\Pr(s_i \to_t s_j|\mathbf{o}) = .25$
  - Then if the Markov process generates $\mathbf{o}$ 100 times, we expect it to see s_i
  - Equivalently, we take $\Pr(s_i \to_t s_j|\mathbf{o})$ as a fractional count
- Sum across time positions

$$f(s_i \to s_j|\mathbf{o}) = \sum_t \Pr(s_i \to_t s_j|\mathbf{o})$$

- Use same re-estimation formulae as for parsed corpus

$$\begin{aligned} a_{ij} &= \Pr(Q_{t+1} = s_j|Q_t = s_i) \hat{=} \frac{f(Q_t = s_i, Q_{t+1} = s_j)}{f(Q_t = s_i)} \\ b_i(w) &= \Pr(\mathbf{o}_t = w|Q_t = s_i) \hat{=} \frac{f(Q_t = s_i, \mathbf{o}_t = w)}{f(Q_t = s_i)} \end{aligned}$$

Iteration

- To compute $\Pr(s_i \to s_j|\mathbf{o})$, etc., we need initial guess

$$M_0 = (a_0, b_0, \pi_0)$$

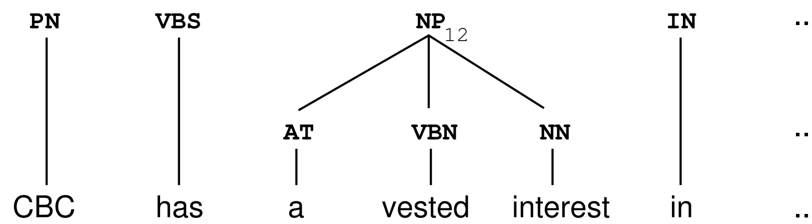
- Iterate using fractional counts to get M_{i+1} from M_i
- Likelihood of model

$$\mathrm{L}(M) = \Pr(\mathbf{o}, M) = \sum_{\mathbf{q}} \Pr(\mathbf{q}, \mathbf{o}, M)$$

- It can be shown that

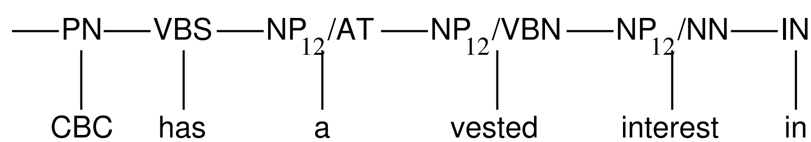
$$\mathrm{L}(M_{i+1}) \geq \mathrm{L}(M_i)$$

- But:
  - Local maximum
  - Overtraining



$$\vdots \qquad \vdots$$
$$\mathbf{NP}_{11} \rightarrow \mathbf{AT\ JJ\ NN}$$
$$\mathbf{NP}_{12} \rightarrow \mathbf{AT\ VBN\ NN}$$
$$\mathbf{NP}_{13} \rightarrow \mathbf{AT\ JJ\ NN\ NN}$$
$$\vdots \qquad \vdots$$

Can be mapped to a standard HMM:



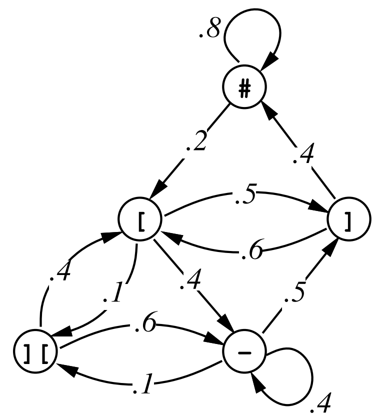
Extensions

- Could also 'tie' states
  - E.g. set $b_{\mathrm{NP}_{12}/\mathrm{AT}} = b_{\mathrm{AT}}$
  - Estimate

$$\hat{b}_{\mathrm{NP}_{12}/\mathrm{AT}}(w) = \hat{b}_{\mathrm{AT}}(w) = \frac{f(\mathrm{NP}_{12}/\mathrm{AT}, w) + f(\mathrm{AT}, w)}{\sum_{w'}[f(\mathrm{NP}_{12}/\mathrm{AT}, w') + f(\mathrm{AT},}$$

- Generalizing to categories other than NP
- Leads to: finite-state chunks

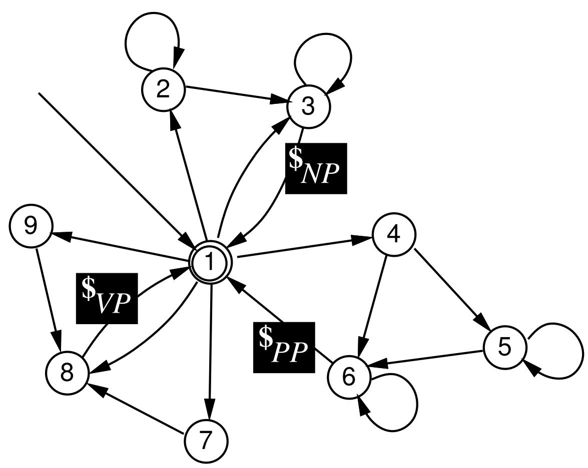
An HMM is a (stochastic) FSA



	[	]	][	-	#
[		.5	.1	.4	
]	.6				.4
][		.4		.6	
-		.5	.1	.4	
#	.2				.8

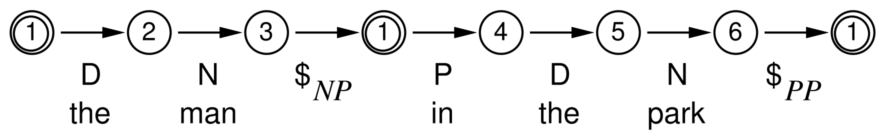
Composing FSA's

$$
\begin{array}{lcl}
\mathrm{NP} & = & \mathrm{D?\ Adj}{*}\ \mathrm{N}{+}\ \boxed{\$_{NP}} \\
\mathrm{PP} & = & \mathrm{P\ NP}\ \boxed{\$_{PP}} \\
\mathrm{VP} & = & (\mathrm{V}\ |\mathrm{Hv\ Vbn}\ |\mathrm{Be\ Vbg})\ \boxed{\$_{VP}} \\
\mathrm{Chunk} & = & \mathrm{NP}\ |\mathrm{PP}\ |\mathrm{VP} \\
\mathrm{S} & \rightarrow & \mathrm{Chunk}{+}
\end{array}
$$

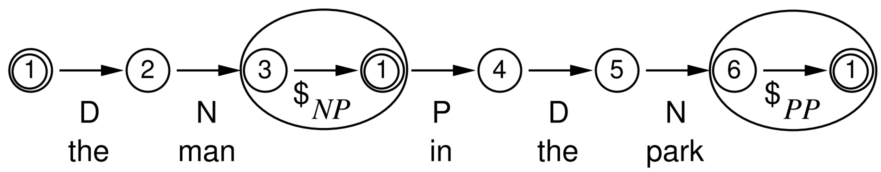


Matching

- Works great if the $'s are in the input

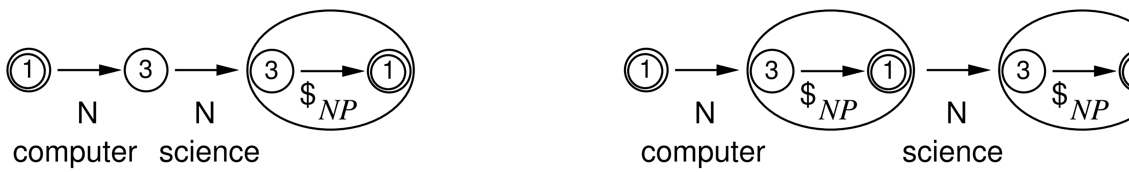


- Fold $'s into surrounding states



Result

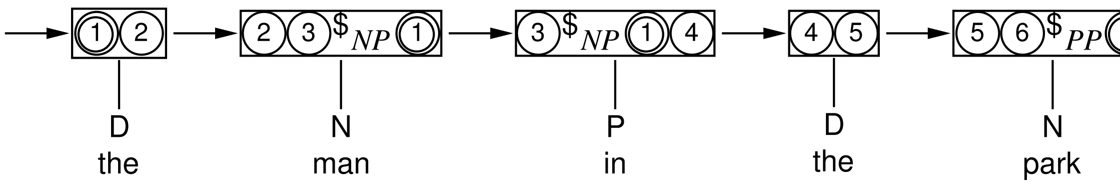
- Add new states $3\$_{NP}1$, $6\$_{NP}1$, $8\$_{NP}1$
- Tie transitions to transitions from original state 1
- Now non-deterministic



- Parse is uniquely recoverable from state-sequence

Final step

- FSA scans on arcs, HMM emits on states
- Turn state-pairs into states

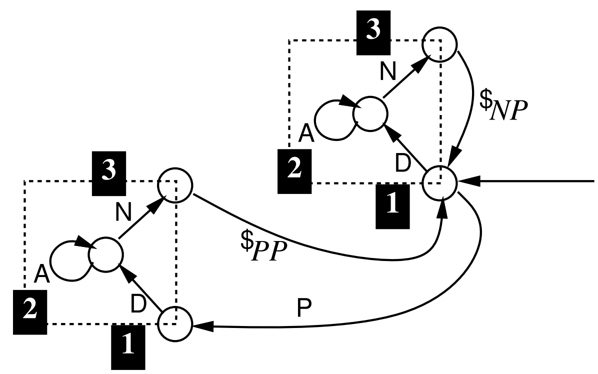


- Transition from $\boxed{ij}$ to $\boxed{jk}$ corresponds to transition from j to k in the unc
- Initial probability of $\boxed{1i}$ represents probability of transition from initial sta

Cascaded FSA's

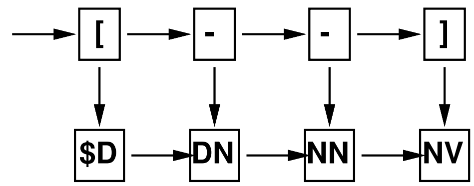
- More of the same medicine

  Clause → PP∗ NP PP∗ VP NP? PP∗ . $\$_{Clause}$
- Insert a copy of the PP regex at each place there's a PP
- Build a large FSA from the resulting regex
- Tie corresponding transitions in different copies of sub-regex

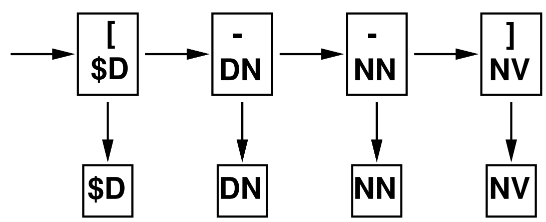


Multiple information sources

- Suppose choice of bracket depends on preceding bracket and preceding tag



- Remember, we cannot do: $\Pr(\mathbf{o}_{t+1}|\mathbf{q}_{t+1}, \mathbf{o}_t) = \Pr(\mathbf{o}_{t+1}|\mathbf{q}_{t+1}) \Pr(\mathbf{o}_{t-}$
- We must estimate the entire distribution $\Pr(\mathbf{o}_{t+1}, \mathbf{q}_{t+1}, \mathbf{o}_t)$
- In effect, we must fold together all information sources into single state

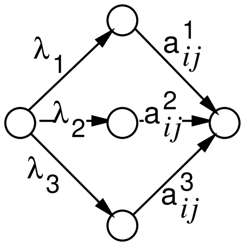


Also for phrase spotting

- Integrate multiple info sources in estimating a_{ij}, $b_i(w)$
  - Folding info sources together leads to state-space explosion, sparse data
  - Combine information from features of state to estimate transition/emissi
- Integrate multiple info sources in estimating precision of phrase-spotting pa
  - Longest match vs. longer-same-cat vs. longer-other-cat vs. overlapping
  - Collocation score
  - Tagging score
  - Phrase type
  - Etc.

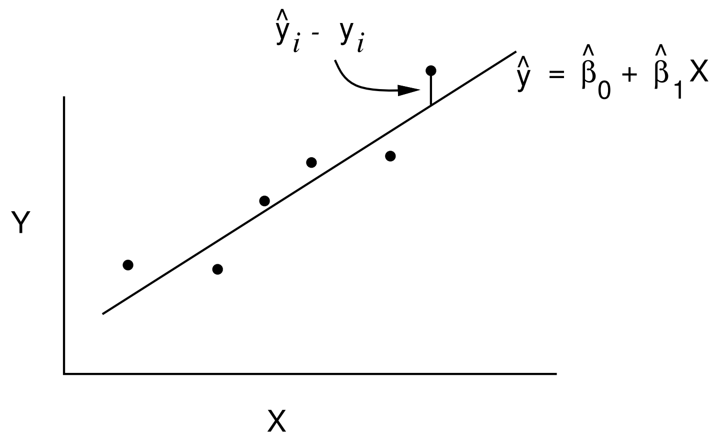
Linear interpolation

- Separately train submodels $M_1, M_2, \ldots$
- E.g., M_1 is an HMM that only looks at previous bracket, and M_2 looks o
tag
- Combine into single model
  - Hold a^k_{ij} fixed
  - Train λ_k
  - Transition probability in combined HMM is $\sum_k \lambda_k k_{ij} = \sum_k \Pr(M_k) \Pr$



Regression

- "Regression analysis is the part of statistics that deals with investigation of the relationship between two or more variables related in a nondeterministic fashion" [68]
- For example: linear regression



$$
\begin{aligned}
Y &= \beta_0 + \beta_1 X + \epsilon \\
\hat{y} &= \hat{\beta}_0 + \hat{\beta}_1 X
\end{aligned}
$$

- Estimating β_0, β_1: minimize squared error $\sum(\hat{y} - y)^2$
- Minimum can be determined analytically from observed pairs (x_i, y_i)
- For given value x, we have point estimate $\hat{y}$ and probability distribution $\hat{p}($

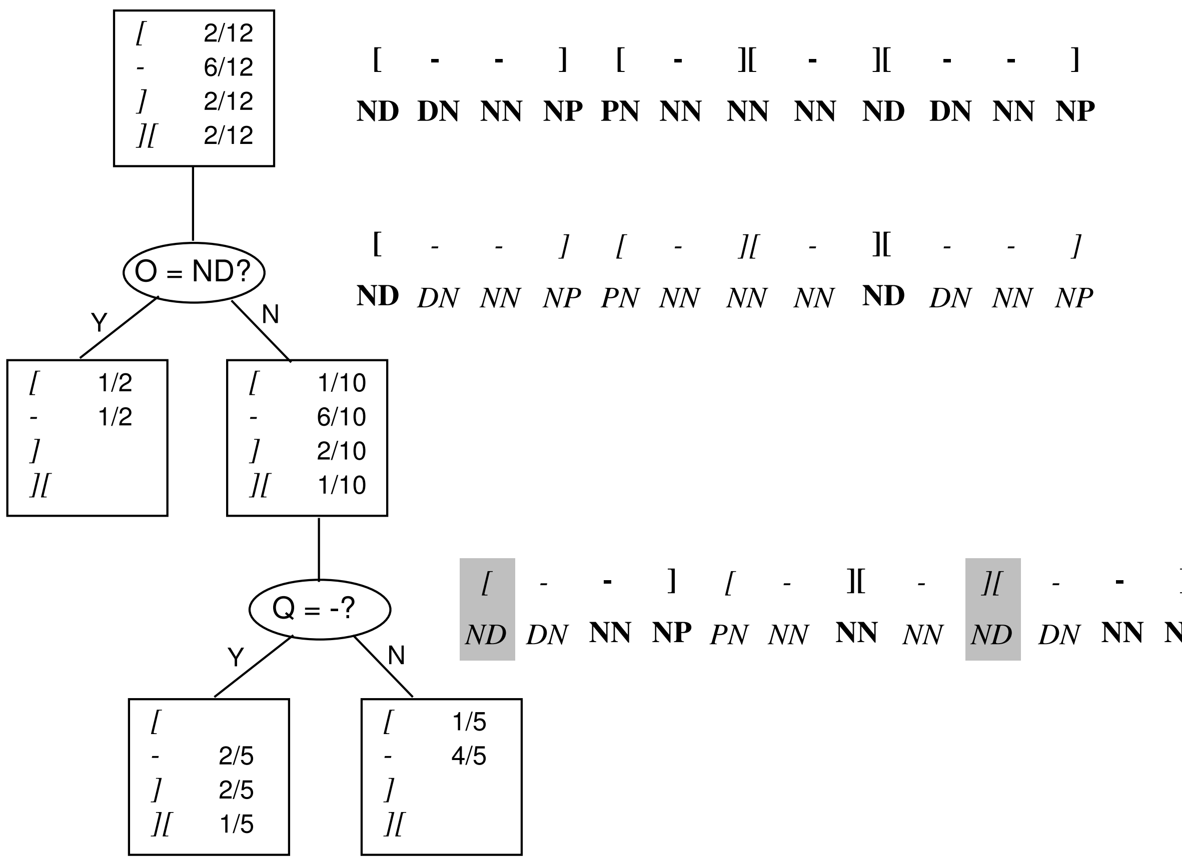
Multivariate regression

- Combining info from multiple variables

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n + \epsilon$$

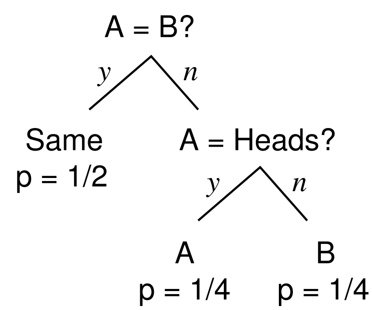
- X_i are *predictor variables*
- Estimate β_i by minimizing squared error
- To do so, need observations $(x_{1i}, \ldots, x_{ni}, y_i)$
- For given values $\langle x_1, \ldots, x_n \rangle$ of predictor variables, we have point estimate a for Y
- Only useful if relationship is approximately linear (though polynomial ger exist)

Regression trees [38, 20]



How good is a split?

- We want to reduce uncertainty about dependent variable
- Uncertainty = entropy
- 1 bit = the uncertainty in one equally-likely two-way guess
- E.g. flip two coins: Same, A, B



Entropy

- Point entropy η – number of 2-way choices to reach given result

$$\begin{aligned} \eta(\text{Same}) &= 1 \\ \eta(\text{A}) &= 2 \\ \eta(\text{B}) &= 2 \end{aligned}$$

- Probability p of ending up at result

$$\begin{aligned} p(\text{Same}) &= 1/2 \\ p(\text{A}) &= 1/4 \\ p(\text{B}) &= 1/4 \end{aligned}$$

- Entropy is average number of 2-way choices = weighted average of η

$$\begin{aligned} &= p(\text{Same})\eta(\text{Same}) + p(\text{A})\eta(\text{A}) + p(\text{B})\eta(\text{B}) \\ &= \tfrac{1}{2} \cdot 1 + \tfrac{1}{4} \cdot 2 + \tfrac{1}{4} \cdot 2 \\ &= 1.5 \end{aligned}$$

Entropy

- In binary-branching tree of uniform depth η containing N leaves

$$\begin{aligned} N &= 2^\eta \\ p &= \frac{1}{N} \end{aligned} \qquad \text{i.e.,} \qquad \begin{aligned} \eta &= \log_2 N \\ N &= \frac{1}{p} \\ \eta &= \log_2 \tfrac{1}{p} \end{aligned}$$

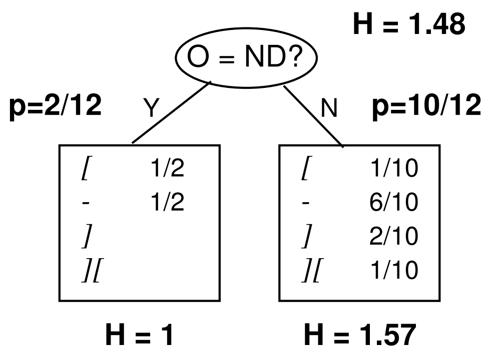
- The same relation can be used generally

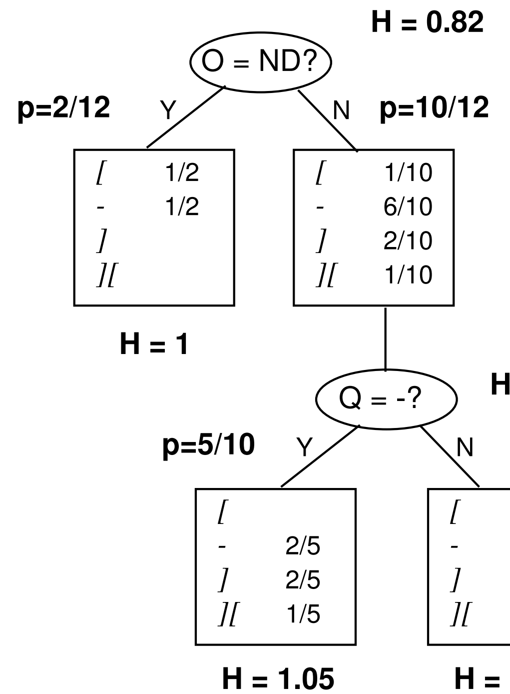
$$\eta_i = \log_2 \frac{1}{p_i} \qquad H = \sum_i p_i \eta_i$$

- Entropy is maximized when all choices are equally likely (maximum uncert
- The more skewed the distribution, the lower the entropy, the lower the unc

Goodness of split

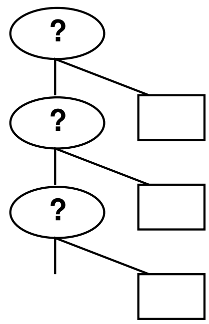
- Goodness of split is reduction in uncertainty: 1.48 - 0.82 = 0.66





Decision lists

- Binary decision tree in which one daughter of every node is a leaf



- Alternative to greedy algorithm (Yarowsky [198])
  - Discriminator: question + answer (Y/N)
  - Evaluate each discriminator independently on *all* data
  - Goodness of discriminator is inverse to uncertainty of resulting leaf dist
  - Sort discriminators by goodness to create decision list

Transformation-based regression (Brill [41])

- Initial assignment rules

  E.g., assign most frequent bracket to tag-pairs
- Error-correction rules $\quad Y \to Y' \ / \ X_1 = x_1, \ldots, X_n = x_n$
- Predictor variables: $X_1, \ldots, X_n$ and Y
- Dependent variable: $Y' = Y$ at $t+1$

- Iterate
  - Evaluate all potential rules
  - Choose best (greedy)
  - Apply, creating a new corpus

- Evaluation
  - Reduction in error rate
  - Errors in corpus after ap
    nus errors before applyi

- Like decision lists, trains on all data
- Only gives point estimate, not distribution

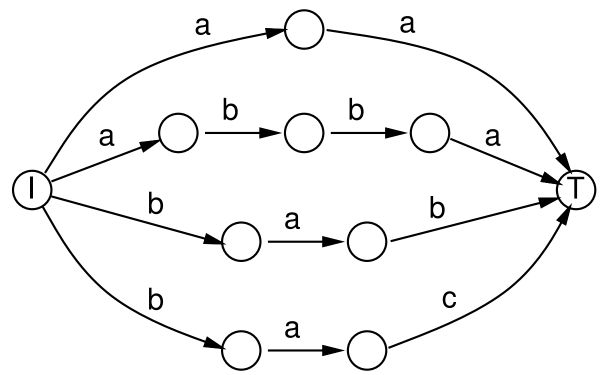
Summary

- User identifies relevant attributes (predictor variables)
- Automatic search through space of discriminators (boolean combinations of predictor variables)
- Point estimate and probability distribution
- State = set of values for predictor variables
- Discriminator = set of states

Grammatical inference: Regular grammars

- Canonical grammar exactly generates training corpus

aa
abba
bab
bac



Bayesian approach [183]

- Prior and posterior
  - Canonical grammar has perfect fit to data
    Highest conditional probability $\Pr(\mathbf{o}|G)$
  - Canonical grammar generally is overly complex
    Low prior probability $\Pr(G)$
  - Likelihood is posterior probability $\Pr(\mathbf{o}, G) = \Pr(\mathbf{o}|G)\Pr(G)$
  - Search for maximum-likelihood grammar
- Operation on grammar: merge two states into one
- Greedy search
  - Consider each pair of states
  - Compute posterior probability if we merge this pair
  - Choose best pair, merge, iterate
  - Quit if no pair improves likelihood

Context-free grammars

- Canonical grammar: one production for each sentence

$$\begin{aligned} S &\rightarrow \text{sentence}_1 \\ S &\rightarrow \text{sentence}_2 \\ &\vdots \end{aligned}$$

- Operators
  - Merge nonterminals
  - Structuring

    Substitute (new) nonterminal X everywhere for sequence $Y_1, \ldots, Y_n$

    Add new rule $X \rightarrow Y_1, \ldots, Y_n$

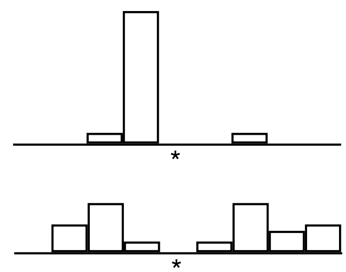
Infering partial grammars: collocations

- Chuch, Gale, Hanks & Hindle [60]
  - Use MI to induce ∼selectional restrictions

    drink ___: ⟨Qty⟩ beer, tea, Pepsi, champagne, liquid, . . .
  - Preprocess with Fidditch to find head-head pairs
- Smadja [177, 176]
  - Use strength of association ∼ MI
  - Also use entropy of positional distribution

    doctor: honorary

    nurse

    
  - Postprocess with Cass

Finch [79]

- Word distribution vectors

	a__	aardvark__	...	__zoologic	__zygote
fish	216	0	...	0	2
habitat	1	5	...	0	0

- Measures of vector (dis)similarity

  Manhattan, Euclidean, dot product, cosine, correlation, rank correlation, divergen

- Cluster words using one of the distance metrics to form parts of speech
- Compute distribution vectors for part-of-speech sequences
- Cluster part-of-speech sequences to form phrase classes

  E.g. 'NP': **C8** (*it*), **C8 C3** (*her status*), **C1 C91 C3** (*the following section*), ...

Smith & Witten [180]

- Special role for function words
- Identify function words by high frequency
  - Another way: bursty → content word (Gale, p.c.; Jones & Sinclair [122]

- Cluster function words

F0: *a, an, her, his, ...*
F1: *he, I, she, then, ...*
F2: *are, be, had, has, ...*

- Form chinks & chunks

F0	C	C	C	F7	F0	C
a	tiny	bird	sat	in	the	tree

- Collect content-word contexts

tiny: F0 __ C C F7
bird: F0 C __ C F7

- Cluster contexts to form content-word categories

F0	C24	C51	C40	F7	F0	C24	C51
a	tiny	bird	sat	in	a	hollow	tree

- Build chink & chunk grammar

$$\begin{aligned} \text{FP0} &\rightarrow \text{F0 C24 C51 C40 F7} \\ \text{FP1} &\rightarrow \text{F7 F0} \\ \text{FP2} &\rightarrow \text{F0 C24 C51 F\$} \end{aligned}$$

- Generalize using substitution operator

$$\text{CP1} \rightarrow \text{C24 C51}$$

American structuralists

- Two measures of phrasehood
  - Substitution (distributional similarity)
  - Cohesiveness
- Substitution

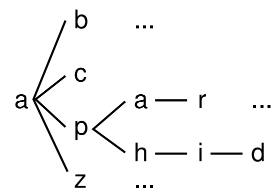
$$he \sim the\ man \qquad \left\{ \begin{array}{l} \_\_laughed \\ \_\_saw\ him \\ he\ saw\ \_\_ \end{array} \right.$$

- Also used by Brill to induce trees
- Current information-theoretic instantiation:
  - Substitution = divergence
  - Cohesiveness = mutual information

Zellig Harris [99]

- American structuralist
  - Sought objective, operational definitions for linguistic concepts
  - Phoneme, morpheme, word, phrase
- “From phoneme to morpheme” [99]
  - Look at number possible continuations for a word prefix

    ap__ {
    a(rtm
    e(rtu
    h(id)
    ⋮
  - Within morpheme, number of possible continuations decreases because



  - Jumps back up at boundary

Example

	h__	he__	hes__	hesc__	hescl__	hescle__	hesclev__	hescleve__
a	and	al	afraid	alm	ad	an		
b		built	bad					
c		came	clever					
d		dgo	dead					
e	elp	ehee	ells	entered	ever		er	
f		ft	orit			ft		
x		xagon	eroxed					
y	ype	y	yodeled	ythed	yde			
z		zoomed	zoomed					
	6	$\boxed{26}$	$\boxed{26}$	9	6	[7]	1	1

Harris

- Do it backwards, too

Agreement	itdisturb__smethatheleft
Cranberry words	cran__berry
Ambiguous prefix	hed__esparatelyneedsit

- Only practical way of getting utterances is elicitation

Stolz 1965 [185]

- Chomsky: "We can be fairly certain that there will be no operational criteria for any but the most elementary [linguistic] notions"
- Seeks operational definition for *phrase* nonetheless
- Phrase = sequence of word-categories co-occuring more frequently than exp
- "Bond"

$$B_{\mathrm{F}}(i) = \frac{\Pr(t_{i+1}|t_1,\ldots,t_i)}{\Pr(t_{i+1})} \qquad B_{\mathrm{B}}(i) = \frac{\Pr(t_{i-1}|t_i,\ldots,t_n)}{\Pr(t_{i-1})}$$

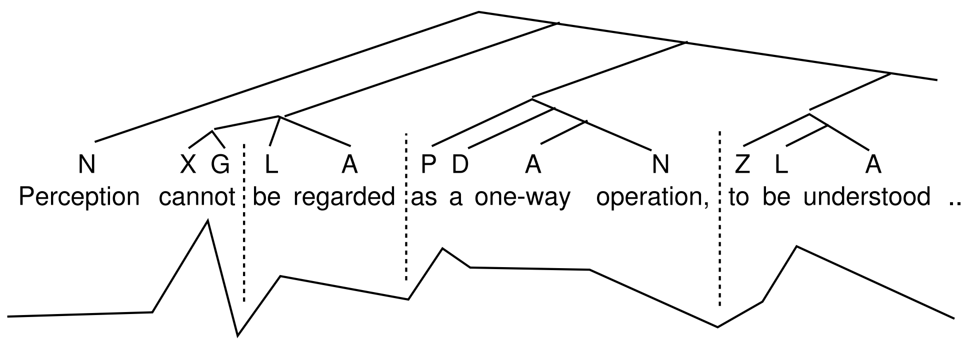
$$B(i) = \tfrac{1}{2}[B_{\mathrm{F}}(i) + B_{\mathrm{B}}(i)]$$

Note: $\log B_{\mathrm{F}}(i) = I(t_{i+1}; t_1, \ldots, t_i)$

- Phrase boundaries at minima in B

Stolz

- Estimates: hand-counted all cat-sequences in a 68,000-word corpus
- Test: 13 sentences from *Scientific American*
- Hand-parsed, differences arbitrated among three judges
- Example



Magerman and Marcus [146]

- Sequences of categories

$$B(i) = \log \frac{\Pr(t_1, \ldots, t_i | t_{i+1}, \ldots, t_n)}{\Pr(t_{i+1})} = \log \frac{\Pr(t_{i+1}, \ldots, t_n | t_1, \ldots}{\Pr(t_1, \ldots, t_i)}$$

- Estimate as product of n-gram MI for windows around i

- Find minimum in window, truncate sentence, repeat

  $\boxed{t_1\ t_2\ t_3\ | t_4}\ t_5\ \ldots$

  $t_1\ t_2\ t_3\ \|\boxed{t_4\ t_5\ | t_6\ t_7}\ t_8\ \ldots$

  – Alternative beginning and end of sentence

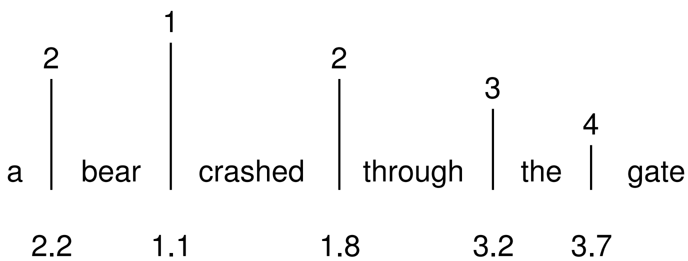
  – Recurse to find constitutents inside these

Bottom line

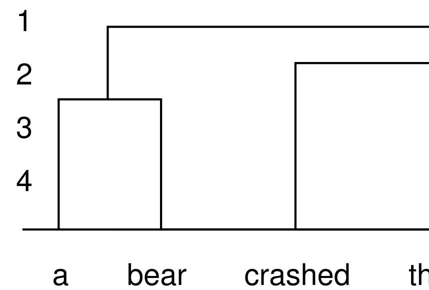
- Works OK for lowlevel phrases
- Important that one use categories, not words
  - Else lexical association pulls phrases apart
    a strong interest_in
  - Function words predict following function words better than following c
    of_the wilderness
  - Result
    an interest_in pictures_of_the Tetons
- Less good at higher levels of structure: here lexical associations are needed

Operational definitions of phrases

- Performance structures
- Naive parsing [96]
  - Subjects divide sentence, redivide



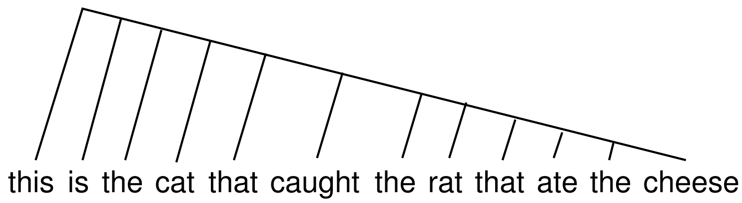
  - Take average prominence of boundary across subjects
  - Dendograms (performar



- Also: transitional error probabilities, pausing, sentence comprehension

Performance structures

- Differ from traditional phrase structures
  - Flatter, no deep right branching
  - Chunk boundaries stable, higher-level boundaries less syntactically pred
- Prosodic phrases differ from traditional phrases in the same way



  - Selkirk: ϕ-phrases [172]
- Gee & Grosjean [92]: use ϕ-phrases to predict performance structures
- Bachenko & Fitzpatrick [18] turn it around and use Gee & Grosjean algor
  intonation for text-to-speech

Linguistics

- The levels sentence, clause, phrase, word are traditional
- Quirk et al. [159] have VP stop at verb

$$[_{NP} \text{ The weather}]\ [_{VP} \text{ has been}]\ [_{AdjP} \text{ remarkably warm}]$$

- Postmodifiers of nouns often assumed Chomsky-adjoined

$$[_{NP}\ [_{NP} \text{ the man}]\ [_{PP} \text{ in the park}]]$$

- Bloch 1946 [31] defines phrases prosodically: "pause-groups"

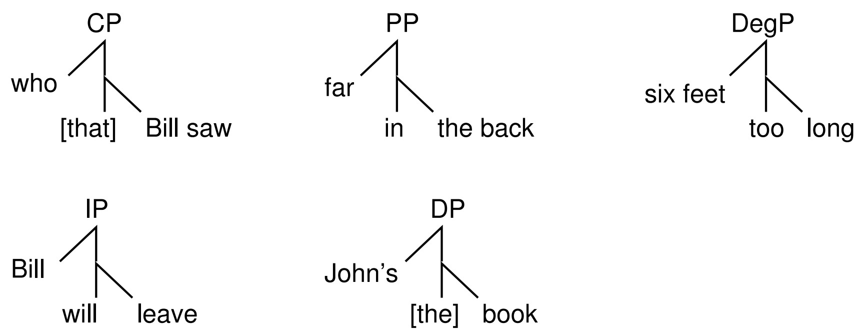
a little dog , with a big bone
*a little , dog with a big , bone

Function Words

- Suzuki (1824)
  - si: noun, verb, adjective – "[si] denotes something"
  - zi: particles – "[zi] denotes nothing; it only attaches 'voice of the heart
- Aristotle
  - Words without meaning: complementizers, conjunctions, etc.
  - Words with meaning: nouns, verbs, adjectives
- Psychology
  - Some aphasias selectively affect function words or content words
  - Slips of the tongue interchange F-F, C-C, but not F-C

Uniform syntactic treatment

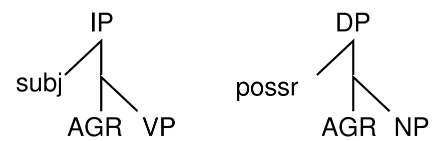
- Function words have subjects and complements [5]



Motivation: Agreement in noun phrase

- English: Tensed verb is *first* verb, not e.g. head:

leaves
was leaving
has been leaving



- Yup'ik: noun phrase has AGR, too

angute-m	kiputaa-Ø	"the man bought it"
angute-t	kiputaa-t	"the men bought it"
angute-m	kuiga-Ø	"the man's river"
angute-t	kuiga-t	"the men's river"

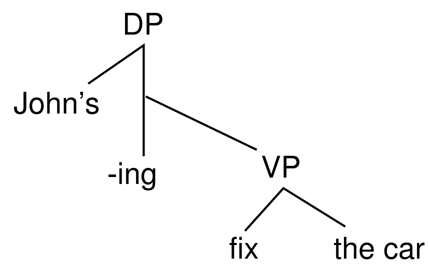
- Turkish

el	"hand"
senin el-in	"your hand"
onun el-i	"his hand"

Motivation: Gerund

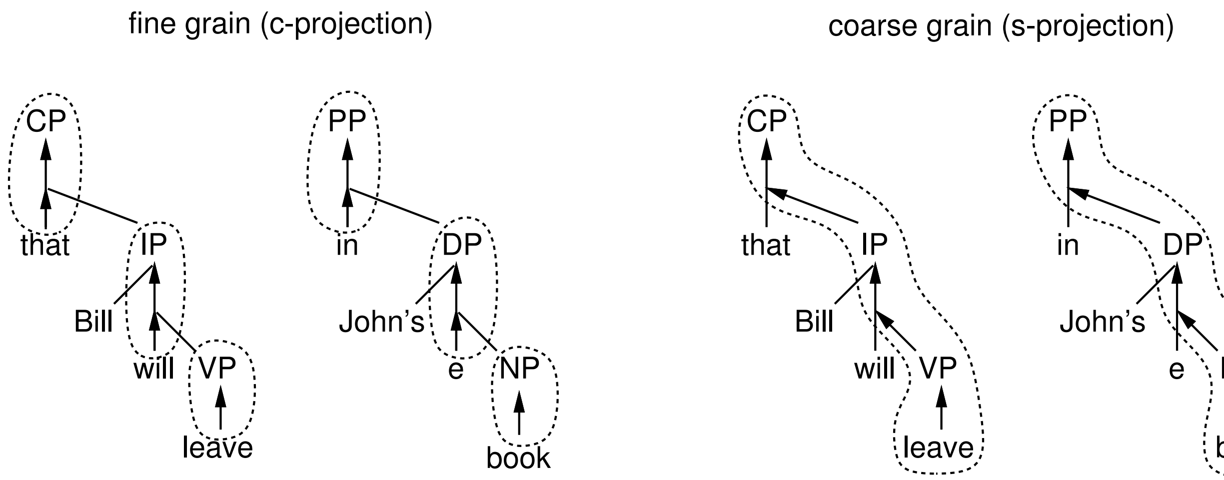
- The Poss-Ing gerund is a gryphon

$$[_{NP} \text{ John's } [_{VP} \text{ fixing the car}]]$$



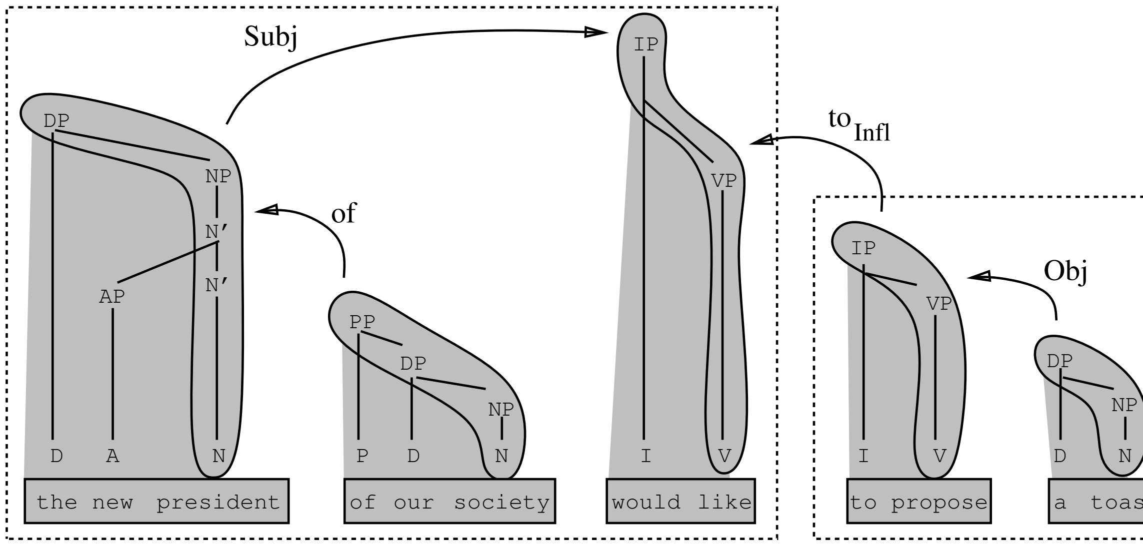
S-projection

- Whether to "count" function words as heads



Chunks and clauses

- Chunk: connected piece of tree covered by an s-projection
- Clause: chunks dominated by same clausal node



Syntactic usefulness of chunks

- No chunk within a chunk [7]

*	[a proud [of his son] man]	[a man] [proud] [of his
*	[a [so tall] man]	[so tall] [a man]
*	[a [six feet] tall man]	[six feet] [tall], [a six-fo
*	[was [every three weeks] fixing] his bike	[was frequently fixing]

- More precisely, F-C selection must be in same chunk

General [2, 3, 4, 35, 36, 50, 61, 62, 81, 82, 84, 116, 117, 118, 129, 143, 144, 148, 2

Tagging [10, 19, 28, 56, 57, 66, 90, 91, 124, 125, 126, 131, 138, 153, 163, 168, 188]

HMMs [21, 22, 23, 24, 25, 49, 64, 67, 78, 115, 119, 155, 157, 160, 161]

Search [156]

The Inside-Outside Algorithm [85, 86, 136, 137]

Regression [20, 30, 29, 38, 41, 42, 45, 46, 154, 162]

Partial Parsing [6, 7, 8, 9, 11, 37, 43, 47, 48, 51, 52, 53, 57, 58, 112, 65, 69, 70, 71, 7
88, 100, 101, 102, 103, 104, 107, 110, 113, 114, 120, 121, 127, 132, 133, 1
147, 149, 152, 163, 164, 165, 166, 169, 178, 182, 186, 190, 191, 192, 194,

Grammatical Inference, Acquisition [1, 12, 13, 14, 15, 16, 32, 33, 39, 40, 55, 58, 7
109, 111, 130, 167, 175, 179, 181, 184, 187, 189, 199]

Mutual Information Parsing [98, 99, 146, 185]

Prosody and Performance Structures [18, 26, 27, 31, 63, 92, 96, 97, 105, 106, 141, 1
173, 193]

References

[1] AAAI. *Proceedings on Machine Learning of Natural Language and Ontology*. G
(DFKI), 1991. Spring Symposium. DFKI Publication D-91-09.

[2] AAAI. *Workshop on Statistically-Based NLP Techniques*, July 1992. Workshop
meeting.

[3] AAAI. *Fall Symposium on Probability and Natural Language Processing*, 1993.

[4] Jan Aarts and T. van den Heuvel. Computational tools for the syntactic ana
Linguistics, 23:303–335, 1985.

[5] Steven Abney. *The English Noun Phrase in its Sentential Aspect.* PhD thesis,
MA, 1987.

[6] Steven Abney. Rapid incremental parsing with repair. In *Proceedings of the 6th
ference: Electronic Text Research*, pages 1–9, Waterloo, Ontario, October 19
Waterloo.

[7] Steven Abney. Syntactic affixation and performance structures. In D. Boucha
editors, *Views on Phrase Structure.* Kluwer Academic Publishers, 1990.

[8] Steven Abney. Parsing by chunks. In Robert Berwick, Steven Abney, and Caro
Principle-Based Parsing. Kluwer Academic Publishers, 1991.

[9] Steven Abney. Chunks and dependencies: Bringing processing evidence to bea
Computational Linguistics and the Foundations of Linguistic Theory. CSLI, To a

[10] Eric Akkerman, Hetty Voog–van Zutphen, and Willem Meijs. *A Computerized L
Level Tagging. ASCOT Report 2.* Language and Computers: Studies in Practica
1. Edited by Jan Aarts and Willem Meijs. Rodopi, Amsterdam, 1988.

[11] H. Alshavi. Processing dictionary definitions with phrasal pattern hierachies.
Linguistics, 13:195–202, 1987.

[12] A. Andreewsky, C. F. Fluhr, and F. Debili. Computational learning of semantic
for the generation and automatical analysis of content. *Information Processing*,

[13] Angluin and Smith. Inductive inference: Theory and methods. *ACM Computin*
1983.

[14] D. Angluin. Inductive inference of formal languages from positive data. *Informa*
45:117–135, 1980.

[15] D. Angluin. Learning regular sets from queries and counterexamples. *Informatio*
tion, 75:87–106, 1987.

[16] Peter Anick and James Pustejovsky. An application of lexical semantics to know
from corpora. In *COLING 90, vol. 2*, pages 7–12, 1990.

[17] Damaris Ayuso et al. Bbn: Description of the PLUM system as used for MU
ings, Fourth Message Understanding Conference (MUC-4), pages 169–176, San M
Morgan Kaufmann.

[18] Joan Bachenko and Elizabeth Fitzpatrick. A computational grammar of discourse
phrasing in English. *Computational Linguistics*, 16(3):155–170, 1990.

[19] L. R. Bahl and R. Mercer. Part-of-speech assignment by a statistical decisio
International Symposium on Information Theory, Ronneby, Sweden, 1976.

[20] Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, and Robert L. Mercer. A tree
language model for natural language speech recognition. *IEEE Trans. on Acous*
Signal Processing, pages 507–514, 1991.

[21] L.R. Bahl, F. Jelinek, and R.L. Mercer. A maximum likelihood approach to c
recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-5

[22] L.E. Baum. An inequality and associated maximization technique in statistic
probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972.

[23] L.E. Baum and J.A. Egon. An inequality with applications to statistical estimatio functions of a markov process and to a model for ecology. *Bull. Amer. Meterol.* 1967.

[24] L.E. Baum, T. Petrie, G. Sopules, and N. Weiss. A maximization technique statistical analysis of probabilistic functions of markov chains. *Annals of Mathe* 41:164–171, 1970.

[25] L.E. Baum and G.R. Sell. Growth functions for transformations on manifolds 27(2):211–227, 1968.

[26] John Bear and Patti Price. Prosody, syntax and parsing. In *28th Annual Meeting for Computational Linguistics*, pages 17–22, 1990.

[27] Mary Beckman and Janet Pierrehumbert. Intonational structure in japanese and e *Yearbook*, 3:255–310, 1986.

[28] J. Benello, A. Mackie, and J. Anderson. Syntactic category disambiguation with *Computer Speech and Language*, 3(3), 1989.

[29] Ezra Black, F. Jelinek, J. Lafferty, R. Mercer, and S. Roukos. Decision tree mod labeling of text with parts-of-speech. In *Darpa Workshop on Speech and Natu* Mateo, CA, 1992. Morgan Kaufman.

[30] Ezra Black, Fred Jelinek, John Lafferty, David M. Magerman, Robert Mercer, a Towards history-based grammars: Using richer models for probabilistic parsing *Meeting of the Association for Computational Linguistics*, Columbus, Ohio, 199

[31] Bernard Bloch. Studies in colloquial Japanese II: Syntax. *Language*, 22:200–248,

[32] Blumer, Ehrenfeucht, Haussler, and Warmuth. Learnability and Vapnik-Chervo *Journal of the ACM*, 36(4), October 1989.

[33] B. Boguraev, R. Byrd, J. Klavans, and M. Neff. From structural analysis of lex
semantics in a lexical knowledge base. In Zernik, editor, *Proceedings of the F
Lexical Acquisition Workshop*. IJCAI, Detroit, 1989.

[34] Branimir Boguraev and Ted Briscoe, editors. *Computational Lexicography for N
Processing*. Longman, New York, 1989.

[35] T. Booth. Probabilistic representation of formal languages. In *Tenth Annual IEE
Switching and Automata Theory*, October 1969.

[36] T.L. Booth and R.A. Thompson. Applying probability measures to abstract l
Trans. Comput., C-22:442–450, 1973.

[37] Didier Bourigault. Surface grammatical analysis for the extraction of terminologi
In *COLING-92, Vol. III*, pages 977–981, 1992.

[38] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression T
and Brooks, 1984.

[39] Michael Brent. Semantic classification of verbs from their syntactic contexts. ms

[40] Michael R. Brent. Automatic acquisition of subcategorization frames from untagg
pora. In *Proceedings of the 29th Annual Meeting of the Association for Computat
pages 209–214, 1991.

[41] Eric Brill. *Transformation-Based Learning*. PhD thesis, Univ. of Pennsylvania, 1

[42] Eric Brill. Some advances in transformation-based part of speech tagging. In *Proc
94*, 1994.

[43] Ted Briscoe. Noun phrases are regular: a reply to Professor Sampson. In W. Mei
Linguistics and Beyond. Rodopi, 1987.

[44] Ted Briscoe, Ann Copestake, and Bran Boguraev. Enjoy the paper: Lexical sema ogy. In *COLING-90, vol. 2*, pages 42–47, 1990.

[45] P. Brown, V. Della Pietra, S. Della Pietra, and R. Mercer. Class-based n-gram language. IBM internal research report, IBM, Yorktown Heights, New York 105

[46] P. Brown, V. Della Pietra, P. deSouza, J. Lai, and R. Mercer. Class-based n-gram language. *Computational Linguistics*, 18(4):467–480, 1992.

[47] Jaime G. Carbonell and Philip J. Hayes. Recovery strategies for parsing extragram *American Journal of Computational Linguistics*, 9(3–4):123–146, 1983.

[48] Jaime G. Carbonell and Philip J. Hayes. Robust parsing using multiple constructi gies. In Leonard Bolc, editor, *Natural Language Parsing Systems*, pages 1–32. Heidelberg, West Germany, 1987.

[49] R. L. Cave and L. P. Neuwirth. Hidden Markov Models for English. In J. D. *Hidden Markov Models for Speech*. IDA-CRD, Princeton, NJ, October 1980.

[50] Eugene Charniak. *(Statistical NLP)*. MIT Press, 1993.

[51] Y. Chiaramella, B. Defude, M. Bruandet, and D. Kerkouba. Iota: A full text inf system. In *Proc. of ACM ICRDIR*, pages 207–213, 1986.

[52] M. Chitrao and R. Grishman. Statistical parsing of messages. In *Proceedings o and Natural Language Processing*. Morgan Kaufman: New York, 1990.

[53] M. Chodorow and J. Klavans. Locating syntactic patterns in text corpora. Ibm report, IBM, Yorktown Heights, New York 10598, 1990.

[54] Y. Choueka. Looking for needles in a haystack or locating interesting collocatio large textual databases. In *Proceedings of the RIAO-88, 609-623. Cambridge, M*

[55] Y. Choueka, S.T. Klein, and E. Neuwitz. Automatic retrieval of frequent idiomatic expressions in a large corpus. *ALLC Journal*, 4(1):34–38, 1983.

[56] Y. Choueka and S. Luisgnan. Disambiguation by short contexts. *Computers and* 19(3):147–157, 1985.

[57] Kenneth Church. A stochastic parts program and noun phrase parser for unres *Proceedings of the Second Conference on Applied Natural Language Processing* 1988.

[58] Kenneth Church. Stochastic parts program and noun phrase parser for unres *ICASSP 89*, pages 695–698, 1989.

[59] Kenneth Church and William Gale. A comparison of the Enhanced Good-Tu Estimation methods for estimating probabilities of English Bigrams. *Compu Language*, 5, 1991.

[60] Kenneth Church, William Gale, Patrick Hanks, and Donald Hindle. Parsing, word typical predicate-argument relations. In *International Workshop on Parsing Te* 389–98, 1989.

[61] Kenneth Church and Robert Mercer. Introduction to the special issue on computa using large corpora. *Computational Linguistics*, 19(1):1–24, 1993.

[62] Computational linguistics **19**(1–2): Special issue on using large corpora, 1993.

[63] W. Cooper and J. Paccia-Cooper. *Syntax and speech.* Harvard University Press, 1980.

[64] M. Cravero, L. Fissore, R. Pieraccini, and C. Scagliola. Syntax driven recognit words by markov models. In *ICASSP 84*, 1984. ¡Stochastic Parsing¿.

[65] Carl G. de Marcken. Parsing the LOB corpus. In *ACL 28*, pages 243–251, 1990.

[66] S. DeRose. Grammatical category disambiguation by statistical optimization. *Linguistics*, 14(1), 1988.

[67] A.-M. Deroualt. Context-dependent phonetic Markov models for large vocabular tion. *Proc. IEEE ICASSP*, 1:360–363, 1987.

[68] Jay L. Devore. *Probability and Statistics for Engineering and the Sciences*. Brooks Co., Monterey, CA, 2nd edition edition, 1987.

[69] M. Dillon and A. Gray. Fasit: A fully automatic syntactically based indexing 34(2):99–108, 1983.

[70] Eva Ejerhed. Finding clauses in unrestricted text by finitary and stochastic methoc *of the 2nd Conference on Applied Natural Language Processing.*, Austin, Texas,

[71] Eva Ejerhed and Kenneth Church. Finite state parsing. In Fred Karlsson, edi *the Seventh Scandinavian Conference of Linguistics*, pages 410–432, Hallituskatu Helsinki 10, Finland, 1983. University of Helsinki, Department of General Lingui

[72] D. Evans, K. Ginther-Webster, M. Hart, R. Lefferts, and I. Monarch. Automat selective nlp and first-order thesauri. In *Proc. of RIAO 91 (Barcelona)*, pages 62

[73] David Evans. Concept management in text via natural-language processing: T proach. In *Text-Based Intelligent Systems: AAAI Spring Symposium.* AAAI, 199

[74] David A. Evans, Steve K. Henderson, Robert G. Lefferts, and Ira A. Monarch the clarit project. Technical Report CMU-LCL-91-2, Laboratory for Computat Carnegie-Mellon University, November 1991.

[75] J. Fagan. *Experiments in Automatic Phrase Indexing of Document Retrieval: Syntactic and Non-Syntactic Methods*. PhD thesis, Cornell University, Ithaca, N

[76] J. Fagan. The effectiveness of a non-syntactic approach to automatic phrase index
retrieval. *JASIS*, 40(2):115–132, 1989.

[77] Jean Fargues and Adeline Perrin. Synthesizing a large concept hierarchy from fre
In *COLING 90, vol. 2*, pages 112–117, 1990.

[78] J. D. Ferguson, editor. *Hidden Markov Models for Speech.* IDA-CRD, Princet
1980.

[79] Steven Paul Finch. *Finding Structure in Language.* PhD thesis, University of Ed

[80] K. S. Fu. *Syntactic pattern recognition and applications.* Prentice-Hall, Englev
1982.

[81] K.S. Fu. *Syntactic Methods in Pattern Recognition.* Springer-Verlag, New York,

[82] K.S. Fu. *Syntactic Methods in Pattern Recognition.* Academic Press, New York,

[83] K.S. Fu and T.L. Booth. Grammatical inference: Introduction and survey. *IEEE T
Man and Cybernetics*, 5, 1975. In two parts.

[84] K.S. Fu and T. Huang. Stochastic grammars and languages. *Int'l. J. of Comp
1(2):135–170, 1972.

[85] T. Fujisaki. A stochastic approach to sentence parsing. In *ACL 22*, 1984. Also app
'84.

[86] T. Fujisaki, F. Jelinek, J. Cocke, and E. Black. Probabilistic parsing method for
biguation. In *Proceedings of the International Workshop on Parsing Technologie*

[87] Robert P. Futrelle et al. Preprocessing and lexicon design for parsing technical
International Workshop on Parsing Technologies, pages 31–40, 1991.

[88] Salton. G. and C. Buckley. A comparison between statistically and syntacticall
phrases. Report TR89-1027, Cornell University, Dept. of Computer Science, Itha

[89] Haim Gaifman. Dependency systems and phrase-structure systems. *Informat*
pages 304–337, 1965.

[90] R. Garside. The CLAWS word-tagging system. In Garside R., F. Leech, and G. S
The Computational Analysis of English. Longman, 1987.

[91] R. Garside, F. Leech, and G. Sampson, editors. *The Computational Analysis of E*
1987.

[92] James Paul Gee and Fran cois Grosjean. Performance structures: A psycholinguis
appraisal. *Cognitive Psychology*, 15:411–458, 1983.

[93] Lila Gleitman. The structural sources of verb meanings. *Language Acquisition*, 1

[94] Ralph Grishman, Lynette Hirschman, and Ngo Thanh Nhan. Discovery procedure
selectional patterns: Initial experiments. *Computational Linguistics*, 12(3), 1986.

[95] Ralph Grishman and John Sterling. Acquisition of selectional patterns. In *C*
658–664, 1992.

[96] F. Grosjean, L. Grosjean, and H. Lane. The patterns of silence: Performance struc
production. *Cognitive Psychology*, 11:58–81, 1979.

[97] Michael Halliday. *Intonation and Grammar in British English*. Mouton, The Ha

[98] Zellig Harris. From morpheme to utterance. *Language*, 22, 1946.

[99] Zellig Harris. From phoneme to morpheme. *Language*, 31, 1955.

[100] Donald Hindle. Deterministic parsing of syntactic non-fluencies. In *ACL 21 (MIT*
1983.

[101] Donald Hindle. User manual for Fidditch. Technical Memorandum #7590-142
Laboratory, 1983.

[102] Donald Hindle. Acquiring disambiguation rules from text. In *Proceedings of*
Meeting of the Assocation of Computational Linguistics, Vancouver, British Colu

[103] Donald Hindle. Noun classification from predicate-argument structures. In *F*
28th Annual Meeting of the Assocation of Computational Linguistics, Pittsburg
268–275, 1990.

[104] Donald Hindle. A parser for text corpora. In A. Zampolli, editor, *Computation*
the Lexicon. Oxford University Press, New York, 1994.

[105] Julia Hirschberg. Prosody bibliography. E-mail, 1986.

[106] Julia Hirschberg. Using text analysis to predict intonational boundaries. Manus
Laboratories, 1991.

[107] Jerry R. Hobbs, Douglas Appelt, Mabry Tyson, and Megumi Kameyama. Fast
extracting information from text. In *ARPA Workshop on Human Language Techn*
CA, 1993. Defense Advanced Research Projects Agency (DARPA), Morgan Kau

[108] Jerry R. Hobss et al. SRI International: Description of the FASTUS system use
Proceedings, Fourth Message Understanding Conference (MUC-4), pages 268–275
1992. Morgan Kaufmann.

[109] James Jay Horning. *A Study of Grammatical Inference*. PhD thesis, Stanford (Co
1969.

[110] Kuang hua Chen and Hsin-Hsi Chen. Extracting noun phrases from large-scale
approach and its automatic evaluation. In *Proceedings of ACL*, 1994. Available
Archive.

[111] Institute of Electrical Engineers and Institute of Mathematics, University of Ess
Inference: Theory, Applications and Alternatives, Colchester, UK, 1993. IEE Pu
no. 1993/092.

[112] J. of computational linguistics **9**(3–4): Special issue on dealing with ill-formed te

[113] Ajay N. Jain. Parsing complex sentences with structured connectionist networks
tation, 3:110–120, 1990.

[114] Ajay N. Jain. *PARSEC: A Connectionist Learning Architecture for Parsing Spoke*
thesis, CMU, Pittsburgh, PA, 1991. Available as Technical Report CMU-CS-91-

[115] F. Jelinek. A fast sequential decoding algorithm using a stack. *IBM Journal*
Development, 13:675–685, November 1969.

[116] F. Jelinek. Self-organized language modeling for speech recognition. *W & L*, pag

[117] F. Jelinek. Self-organized language modeling for speech recognition. In *IBM Euro*
Advances in Speech Recognition (Oberlech, Austria), 1986.

[118] F. Jelinek, J. D. Lafferty, and R. L. Mercer. Basic methods of probabilistic contex
Research Report RC 16374 (#72684), IBM, Yorktown Heights, New York 10598,

[119] F. Jelinek and R. Mercer. Interpolated estimation of Markov source parameters fro
E.S Gelsema and Kanal L.N., editors, *Pattern Recognition in Practice*, pages 381-
: North Holland Publishing Co., 1980.

[120] K. Jensen and G.E. Heidorn. The fitted parse: 100English. Computer Science
RC9729 (#42958), IBM Research Division (San Jose), 1982.

[121] K. Jensen, G.E. Heidorn, L.A. Miller, and Y. Ravin. Parse fitting and prose fixin
of ill-formedness. *Computational Linguistics*, 9(3–4):147–161, 1983.

[122] S. Jones and J. Sinclair. English lexical collocations: A study in Computational Li
de Lexicologie, 24:15–49, 1974.

[123] Aravind K. Joshi and B. Srinivas. Disambiguation of super parts of speech (or su
parsing. In *COLING-94*, 1994.

[124] Gunnel Kællgren. Tagging pilys 47. Technical report, Institute of Linguistics, Un
holm, 1982.

[125] Gunnel Kællgren. Making maximal use of surface criteria in large-scale parsing:
unpublished ms., 1990.

[126] Fred Karlsson. Morphological tagging of Finnish. In *Computational Morphosy
No. 13*, pages 115–136. University of Helsinki, Department of General Linguistic

[127] Fred Karlsson. Parsing and constraint grammar. unpublished ms., Research Unit f
Linguistics, Helsinki, Finland, 1989.

[128] Judith Klavans. Complex: a computational lexicon for natural language systems
1988.

[129] Judith L Klavans. Bibliography on corpus analysis and tagging. presented at th
role of large text corpora in building natural language systems at the 13th interna
on computational linguistics (coling), 1990.

[130] Julian Kupiec. Training stochastic grammars from unlabelled text corpora. Ms.,

[131] Julian Kupiec. Augmenting a hidden Markov model for phrase-dependent word
ceedings of the DARPA Speech and Natural Language Workshop. Morgan Kau
1989.

[132] Julian Kupiec. An algorithm for finding noun phrase correspondences in bilin
ACL, pages 17–22, 1993.

[133] Stan Kwasny and N. Sondheimer. Relaxation techniques for parsing ill-forme
7(2):99–108, 1981.

[134] Lang. Parsing incomplete sentences. In *Proceedings of COLING '88*, 1988.

[135] Francois-Michel Lang and Lynette Hirschman. Improved portability and parsin
active acquisition of semantic information. In *Proceedings of the Second Confe
Natural Language Processing*, Austin, TX, 1988. ACL.

[136] K. Lari and S.J. Young. The estimation of stochastic context-free grammars
Outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.

[137] K. Lari and S.J. Young. Applications of stochastic context-free grammars using t
algorithm. *Computer Speech and Language*, 5:237–257, 1991.

[138] Leech, Garside, and Atwell. The automatic grammatical tagging of the lob corpu
7:13–33, 1983.

[139] Wendy Lehnert et al. University of Massachusetts: MUC-4 test results and ana
ings, Fourth Message Understanding Conference (MUC-4), pages 151–158, San
Morgan Kaufmann.

[140] L. Lesmo and P. Torasso. Interpreting syntactically ill-formed sentences. In *CO*

[141] W.J.M. Levelt. Hierarchial chunking in sentence processing. *Perception & Psych*
103, 1970.

[142] D. D. Lewis and W. B. Croft. Term clustering of syntactic phrases. In *SIGIR '9*
1990.

[143] Mark Liberman. Tutorial: Statistical methods in nl processing. In *EACL-93*, 19

[144] Mark Liberman and Mitch Marcus. (statistical nlp). *CACM*, 1994?

[145] David D. MacDonald. An efficient chart-based algorithm for partial parsing of u
In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, 1

[146] D. Magerman and M. Marcus. Parsing a natural language using mutual informa
Proceedings of AAAI-90, 1990.

[147] Christopher D. Manning. Automatic acquisition of a large subcategorization dic
pora. In *31st Annual Meeting of the Association for Computational Linguistic*
1993.

[148] Mitch Marcus. (statistical nlp). Tutorial, ACL 1992, 1992.

[149] Mitchell Marcus. Building non-normative systems: the search for robustness: an
20, page 152, 1982.

[150] Mitchell P. Marcus, Donald Hindle, and Margaret M. Fleck. D-theory: Talking ab
trees. Manuscript, Bell Laboratories.

[151] James G. Martin. Rhythmic (hierarchical) versus serial structure in speech an
Psychological Review, 79(6):487–509, 1972.

[152] Chris S. Mellish. Some chart-based techniques for parsing ill-formed input. In *P*
'89, 1989.

[153] Meteer, Schwartz, and Weischedel. Studies in part of speech labelling. In *Proceedi*
Speech and Natural Language Workshop. Morgan Kaufmann, 1991.

[154] Frederick Mosteller and John W. Tukey. *Data Analysis and Regression*. Addison-
Company, Reading MA, 1977.

[155] Douglas B. Paul. Speech recognition using Hidden Markov Models. *Lincoln La*
3(1):41–62, 1990.

[156] Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem S*
Wesley Publishing Company, Reading, MA, 1984.

[157] Joseph Picone. Continuous speech recognition using Hidden Markov Models. *IE*
zine, pages 26–41, July 1990.

[158] James Pustejovsky, Sabine Bergler, and Peter Anick. Lexical semantic technique
ysis. Ms., Brandeis, 1992.

[159] R. Quirk, S. Greenbaum, G. Leech, and J. Svartik. *A Comprehensive Gramm*
Language. Longman: London, 1985.

[160] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in sp
Proceedings of the IEEE, 77(2):257–285, February 1989.

[161] L.R. Rabiner and B.H. Juang. An introduction to Hidden Markov Models. *IEEE*
page 4ff, January 1986.

[162] Lance A. Ramshaw. Exploring the statistical derivation of transformational r
part-of-speech tagging. In *Proceedings of the ACL Balancing Act Workshop*, 199

[163] Mats Rooth. Unitary stochastic part-of-speech and phrase tagging. Manuscri
Stuttgart, 1994.

[164] Ian C. Ross and John W. Tukey. Introduction to these volumes. In *Index*
Probability, pages iv–x. The R & D Press, Los Altos, CA, 1975.

[165] G. Ruge, C. Schwarz, and A. Warner. Effectiveness and efficiency in natural lan
for large amounts of text. *JASIS*, 42(6):450–456, 1991.

[166] Gerard Salton and Maria Smith. On the application of syntactic methodologies i analysis. In *Proceedings of the 12th Annual International ACM/SIGIR Confer and Development in Information Retrieval*, pages 137–150, 1989.

[167] E. Sanchis, F. Casacuberta, I. Galiano, and E. Segarra. Learning structural m units through grammatical inference. In *IEEE ICASSP, Vol. 1*, pages 189–192, 1

[168] Beatrice Santorini. Part-of-speech tagging guidelines for the Penn Treebank Pr Report MS-CIS-90-47/LINC LAB 178, Dept. of Computer and Information Scie Pennsylvania, Philadelphia, PA, July 1990.

[169] Christoph Schwarz. Automatic syntactic analysis of free text. *JASIS*, 41(6):408–

[170] Elisabeth O. Selkirk. On prosodic structure and its relation to syntactic structure editor, *Nordic Prosody II.* Tapir, Trondheim, 1978.

[171] Elisabeth O. Selkirk. Prosodic domains in phonology: Sanskrit revisited. In M. A Kean, editors, *Juncture*, pages 107–129. Anma Libri, Saratoga, CA, 1980.

[172] Elisabeth O. Selkirk. On the nature of phonological representations. In T. Mye J. Anderson, editors, *The Cognitive Representation of Speech.* North-Holland Pub Amsterdam, 1981.

[173] Elisabeth O. Selkirk. *Phonology and Syntax: The Relation between Sound and* Press, Cambridge, MA, 1984.

[174] Stephanie Seneff. A relaxation method for understanding spontaneous speech ut *ceedings, Speech and Natural Language Workshop*, San Mateo, CA, 1992. DARP mann Publishers.

[175] *Extraction of Hierarchical Structure for Machine Learning of Natural Language* and Language Technology (ITK), 1992. Proceedings 92/1, ISBN 90-74029-02-7.

[176] Frank Smadja. *Extracting Collocations from Text. An Application: Language G*
thesis, Columbia University, New York, NY, 1991.

[177] Frank Smadja and Kathy McKeown. Automatically extracting and representing
language generation. In *Proceedings of the 28th Annual Meeting of the Assocation*
Linguistics, pages 252–259, 1990.

[178] A. Smeaton. Using parsing of natural language as part of document retrieval.
CSC/88/R1, University of Glasgow, 1988.

[179] A.R. Smith et al. Application of a sequential pattern learning system to connec
nition. In *ICASSP '85*, 1985.

[180] Tony C. Smith and Ian H. Witten. Language inference from function words. Manu
of Calgary and University of Waikato, January 1993.

[181] P. Smyth and R.M. Goodman. An information theoretic approach to rule inductio
IEEE Transactions on Knowledge and Data Engineering, 4(4):301–317, August 1

[182] David Stallard and Robert Bobrow. Fragment processing in the DELPHI system
Speech and Natural Language Workshop, pages 305–310, San Mateo, CA, 1992.
Research Projects Agency (DARPA), Morgan Kaufmann Publishers.

[183] Andreas Stolcke and Stephen Omohundro. Inducing probabilistic grammars by
merging. From cmp-lg archive, 1994.

[184] Andreas Stolcke and Stephen M. Omohundro. Best-first model merging for hidd
induction. Technical report, International Computer Science Institute, Berkeley
uary 1994.

[185] W. Stolz. A probabilistic procedure for grouping words into phrases. *Language a*
235, 1965.

[186] Tomek Strzalkowski. Ttp: A fast and robust parser for natural language. In *C*
198–204, 1992.

[187] Stan Szpakowicz. Semi-automatic acquisition of conceptual structure from techn
Man-Machine Studies, 33:385–397, 1990.

[188] E. Tzoukermann and B. Merialdo. Some statistical approaches for tagging unre
published ms., IBM, T. J. Watson Research Center, Yorktown Heights, New Yor

[189] L. G. Valiant. A theory of the learnable. In *Proceedings of the ACM Symposi*
Computing, pages 436–445, Washington, D.C., 1984. ACM Press.

[190] Jacques Vergnes. ?? In *COLING '90*, 1990.

[191] Atro Voutilainen. NPtool, a detector of English noun phrases. In *Proceedings of*
Very Large Corpora, pages 48–57, 1993.

[192] Atro Voutilainen, Juha Heikkilä, and Arto Anttila. Constraint grammar of english
oriented introduction. Technical Report Publication No. 21, University of Helsin
General Linguistics, Helsinki, 1992.

[193] Michelle Q. Wang and Julia Hirschberg. Predicting intonational phrasing from te

[194] Weischedel and Black. Responding intelligently to unparsable inputs. *Amer. J.*
6(2):97–109, 1980.

[195] Ralph Weischedel et al. Partial parsing: A report on work in progress. In *P*
DARPA Speech and Natural Language Workshop, pages 204–209, Asilomar, CA,

[196] R.M. Weischedel and N.K. Sondheimer. Meta-rules as a basis for processing i
Amer. J. of CL, 9:161–177, 1983.

[197] Yorick Wilks, Louise Guthrie, Joe Guthrie, and Jim Cowie. Combining weak m
scale text processing. In Paul S. Jacobs, editor, *Text-Based Intelligent Systems:
and Practice in Information Extraction and Retrieval*, pages 35–58. Lawrence Erl
1992.

[198] David Yarowsky. Decision lists for leixcal ambiguity resolution. Manuscript, Un
sylvania, 1994.

[199] Uri Zernik and Paul Jacobs. Tagging for learning: Collecting thematic relations
COLING '90 vol. 1, pages 34–39, 1990.

[200] Wu Zhibiao. A survey of statistical-based approaches to nlp. ms., 1993.