Statistical NLP

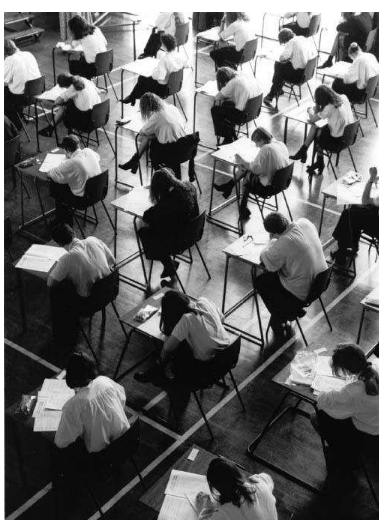
Spring 2010

University of California
CAL NLP
Berkeley

Lecture 15: Grammar Induction

Dan Klein – UC Berkeley

Supervised Learning



- Systems duplicate correct analyses from training data
- Hand-annotation of data
  - Time-consuming
  - Expensive
  - Hard to adapt for new purposes (tasks, languages, domains, etc)
  - Corpus availability drives research, not tasks
- Example: Penn Treebank
  - 50K Sentences
  - Hand-parsed over several years

Unsupervised Learning

- Systems take raw data and automatically detect patterns
- Why unsupervised learning?
  - More data than annotation
  - Insights into machine learning, clustering
  - Kids learn some aspects of language entirely without supervision
- Here: unsupervised learning
  - Work purely from the forms of the utterances
  - Neither assume nor exploit prior meaning or grounding [*cf.* Feldman *et al.*]

Unsupervised Parsing?

- Start with raw text, learn syntactic structure
- Some have argued that learning syntax from positive data alone is impossible:
  - Gold, 1967: Non-identifiability in the limit
  - Chomsky, 1980: The poverty of the stimulus
- Many others have felt it should be possible:
  - Lari and Young, 1990
  - Carroll and Charniak, 1992
  - Alex Clark, 2001
  - Mark Paskin, 2001
  - … and many more, but it didn't work well (or at all) until the past few years
- Surprising result: it's possible to get entirely unsupervised parsing to (reasonably) work well!

Learnability

- Learnability: formal conditions under which a class of languages can be learned in some sense
- Setup:
  - Class of languages is $\mathscr{L}$
  - Learner is some algorithm H
  - Learner sees a sequences X of strings $x_1 \ldots x_n$
  - H maps sequences X to languages L in $\mathscr{L}$
- Question: for what classes do learners exist?

Learnability: [Gold 67]

- Criterion: identification in the limit
  - A **presentation** of L is an infinite sequence of x's from L in which each x occurs at least once
  - A learner H **identifies L in the limit** if for any presentation of L, from some point n onward, H always outputs L
  - A class $\mathscr{L}$ is **identifiable in the limit** if there is some single H which correctly identifies in the limit any L in $\mathscr{L}$
- Example: L = {{a}, {a,b}} is learnable in the limit
- Theorem [Gold 67]: Any $\mathscr{L}$ which contains all finite languages and at least one infinite language (i.e. is superfinite) is unlearnable in this sense

Learnability: [Gold 67]

- Proof sketch
  - Assume $\mathscr{L}$ is superfinite
  - There exists a chain $L_1 \subset L_2 \subset \ldots L_\infty$
  - Take any learner H assumed to identify $\mathscr{L}$
  - Construct the following misleading sequence
    - Present strings from L_1 until it outputs L_1
    - Present strings from L_2 until it outputs L_2
    - …
  - This is a presentation of L_∞, but H won't identify L_∞

Learnability: [Horning 69]

- Problem: IIL requires that H succeed on each presentation, even the weird ones

- Another criterion: **measure one identification**
  - Assume a distribution $P_L(x)$ for each L
  - Assume $P_L(x)$ puts non-zero mass on all and only x in L
  - Assume infinite presentation X drawn i.i.d. from $P_L(x)$
  - H measure-one identifies L if probability of drawing an X from which H identifies L is 1

- [Horning 69]: PCFGs can be identified in this sense
  - Note: there can be misleading sequences, they just have to be (infinitely) unlikely

Learnability: [Horning 69]

- Proof sketch
  - Assume $\mathscr{L}$ is a recursively enumerable set of recursive languages (e.g. the set of PCFGs)
  - Assume an ordering on all strings $x_1 < x_2 < \ldots$
  - Define: two sequences A and B **agree through n** if for all $x < x_n$, x in A $\Leftrightarrow$ x in B
  - Define the **error set** E(L,n,m):
    - All sequences such that the first m elements do not agree with L through n
    - These are the sequences which contain early strings outside of L (can't happen) or fail to contain all the early strings in L (happens less as m increases)
  - Claim: P(E(L,n,m)) goes to 0 as m goes to ∞
  - Let $d_L(n)$ be the smallest m such that $P(E) < 2^{-n}$
  - Let d(n) be the largest $d_L(n)$ in first n languages
  - Learner: after d(n) pick first L that agrees with evidence through n
  - Can only fail for sequence X if X keeps showing up in E(L,n,d(n)), which happens infinitely often with probability zero (we skipped some details)

Learnability

- Gold's result says little about real learners (requirements of IIL are way too strong)
- Horning's algorithm is completely impractical (needs astronomical amounts of data)
- Even measure-one identification doesn't say anything about tree structures (or even density over strings)
  - Only talks about learning grammatical sets
  - Strong generative vs weak generative capacity

Unsupervised Tagging?

- AKA part-of-speech induction
- Task:
  - Raw sentences in
  - Tagged sentences out
- Obvious thing to do:
  - Start with a (mostly) uniform HMM
  - Run EM
  - Inspect results

EM for HMMs: Process

- Alternate between recomputing distributions over hidden variables (the tags) and reestimating parameters
- Crucial step: we want to tally up how many (fractional) counts of each kind of transition and emission we have under current params:

$$\texttt{count}(w, s) = \sum_{i:w_i=w} P(t_i = s|\mathbf{w})$$

$$\texttt{count}(s \rightarrow s') = \sum_{i} P(t_{i-1} = s, t_i = s'|\mathbf{w})$$

- Same quantities we needed to train a CRF!

Merialdo: Setup

- Some (discouraging) experiments [Merialdo 94]
- Setup:
  - You know the set of allowable tags for each word
  - Learn a supervised model on k training sentences
    - Learn P(w|t) on these examples
    - Learn $P(t|t_{-1},t_{-2})$ on these examples
  - On n > k sentences, re-estimate with EM
- Note: we know allowed tags but not frequencies

Merialdo: Results

	Number of tagged sentences used for the initial model						
	0	100	2000	5000	10000	20000	all
Iter	Correct tags (% words) after ML on 1M words						
0	77.0	90.0	95.4	96.2	96.6	96.9	97.0
1	80.5	92.6	95.8	96.3	96.6	96.7	96.8
2	81.8	93.0	95.7	96.1	96.3	96.4	96.4
3	83.0	93.1	95.4	95.8	96.1	96.2	96.2
4	84.0	93.0	95.2	95.5	95.8	96.0	96.0
5	84.8	92.9	95.1	95.4	95.6	95.8	95.8
6	85.3	92.8	94.9	95.2	95.5	95.6	95.7
7	85.8	92.8	94.7	95.1	95.3	95.5	95.5
8	86.1	92.7	94.6	95.0	95.2	95.4	95.4
9	86.3	92.6	94.5	94.9	95.1	95.3	95.3
10	86.6	92.6	94.4	94.8	95.0	95.2	95.2

Distributional Clustering

♦ *the president said that the downturn was over* ♦

president	*the __ of*
president	*the __ said*
governor	*the __ of*
governor	*the __ appointed*
said	*sources __ ♦*
said	*president __ that*
reported	*sources __ ♦*

president governor

said reported

the a

[Finch and Chater 92, Shuetze 93, many others]

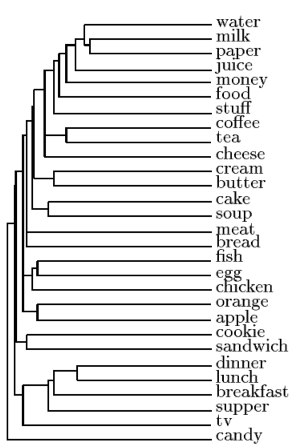
Distributional Clustering

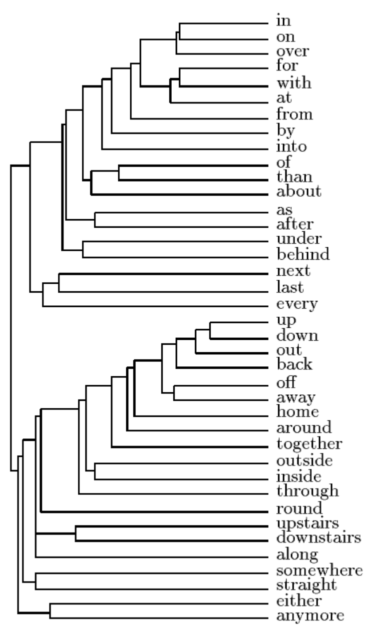
- Three main variants on the same idea:
  - Pairwise similarities and heuristic clustering
    - E.g. [Finch and Chater 92]
    - Produces dendrograms
  - Vector space methods
    - E.g. [Shuetze 93]
    - Models of ambiguity
  - Probabilistic methods
    - Various formulations, e.g. [Lee and Pereira 99]

Nearest Neighbors

word	nearest neighbors
accompanied	submitted banned financed developed authorized headed canceled awarded barred
almost	virtually merely formally fully quite officially just nearly only less
causing	reflecting forcing providing creating producing becoming carrying particularly
classes	elections courses payments losses computers performances violations levels pictures
directors	professionals investigations materials competitors agreements papers transactions
goal	mood roof eye image tool song pool scene gap voice
japanese	chinese iraqi american western arab foreign european federal soviet indian
represent	reveal attend deliver reflect choose contain impose manage establish retain
think	believe wish know realize wonder assume feel say mean bet
york	angeles francisco sox rouge kong diego zone vegas inning layer
on	through in at over into with from for by across
must	might would could cannot will should can may does helps
they	we you i he she nobody who it everybody there

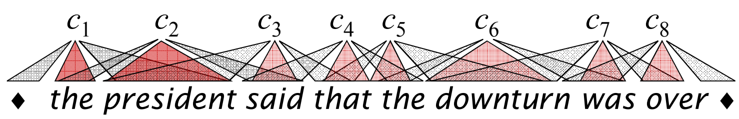
Dendrograms

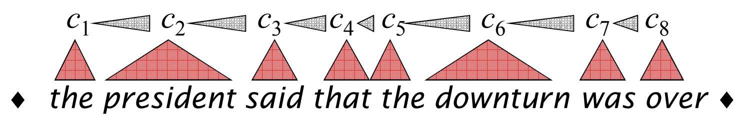




A Probabilistic Version?

$$P(S,C) = \prod_i P(c_i)P(w_i \mid c_i)P(w_{i-1}, w_{i+1} \mid c_i)$$





Weakly Supervised Learning

Newly remodeled 2 Bdrms/1 Bath, spacious upper unit, located in Hilltop Mall area. Walking distance to shopping, public transportation, schools and park. Paid water and garbage. No dogs allowed.

Prototype Lists

FEATURE	kitchen, laundry
LOCATION	near, close
TERMS	paid, utilities
SIZE	large, feet
RESTRICT	cat, smoking

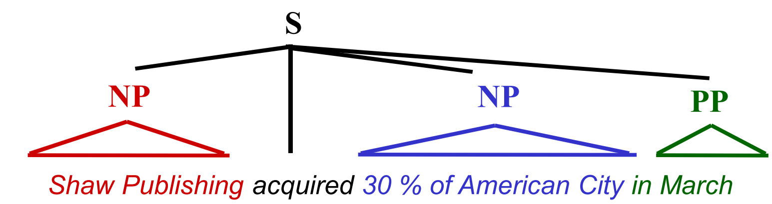
Information Extraction

NN	president	IN	of
VBD	said	NNS	shares
CC	and	TO	to
NNP	Mr.	PUNC	.
JJ	new	CD	million
DET	the	VBP	are

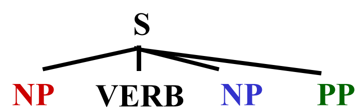
English POS

From [Haghighi and Klein 06]

Context-Free Grammars



- Looks like a context-free grammar.
- Can model a tree as a collection of context-free rewrites (with probabilities attached).



$$P(\text{NP VERB NP PP} \mid \text{S}) = 0.1$$

Early Approaches: Structure Search

- Incremental grammar learning, chunking [Wolff 88, Langley 82, many others]
  - Can recover synthetic grammars
- An (extremely good / lucky) result of incremental structure search:

N-bar or zero determiner NP
zNN → NN | NNS
zNN → JJ zNN
zNN → zNN zNN

NP with determiner
zNP → DT zNN
zNP → PRP$ zNN

Proper NP
zNNP → NNP | NNPS
zNNP → zNNP zNNP

Transitive VPs
(complementation)
zVP → zV JJ
zVP → zV zNP
zVP → zV zNN
zVP → zV zPP

Transitive VPs
(adjunction)
zVP → zRB zVP
ZVP → zVP zPP

PP
zPP → zIN zNN
zPP → zIN zNP
zPP → zIN zNNP

verb groups / intransitive VPs
zV → VBZ | VBD | VBP
zV → MD VB
zV → MD RB VB
zV → zV zRB
zV → zV zVBG

Intransitive S
zS → PRP zV
zS → zNP zV
zS → zNNP zV

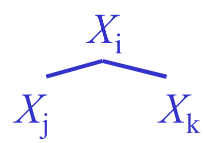
Transitive S
zSt → zNNP zVP
zSt → zNN zVP
zSt → PRP zVP

- Looks good, … but can't parse in the wild.

Idea: Learn PCFGs with EM

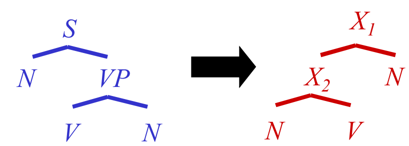
- Classic experiments on learning PCFGs with Expectation-Maximization [Lari and Young, 1990]

$$\{ X_1 , X_2 \ldots X_n \}$$



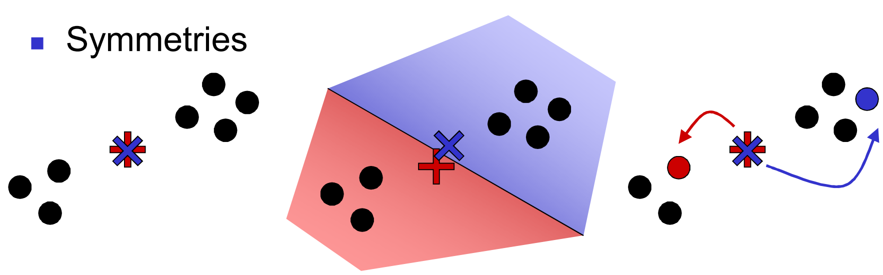
  - Full binary grammar over n symbols
  - Parse uniformly/randomly at first
  - Re-estimate rule expectations off of parses
  - Repeat

- Their conclusion: it doesn't really work.

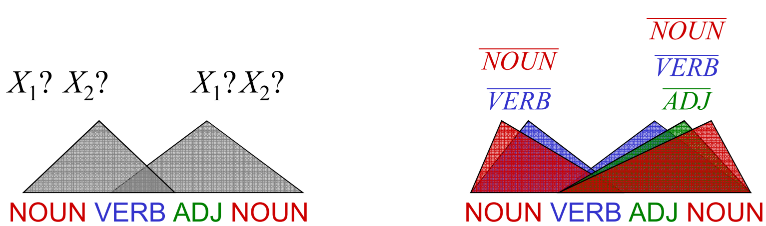


Problem: Model Symmetries

- Symmetries



- How does this relate to trees

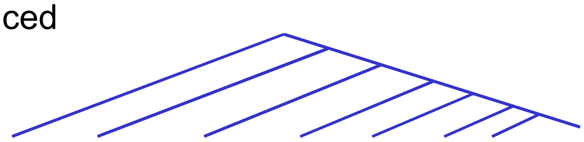


Other Approaches

- Evaluation: fraction of nodes in gold trees correctly posited in proposed trees (unlabeled recall)
- Some recent work in learning constituency:
  - [Adrians, 99] Language grammars aren't general PCFGs
  - [Clark, 01] Mutual-information filters detect constituents, then an MDL-guided search assembles them
  - [van Zaanen, 00] Finds low edit-distance sentence pairs and extracts their differences

Right-Branching Baseline

- English trees tend to be right-branching, not balanced



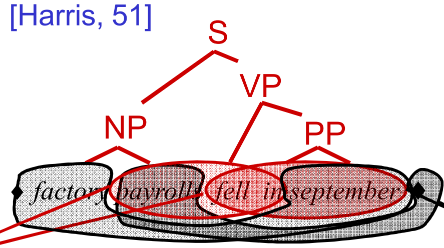
they were unwilling to agree to new terms

- A simple (English-specific) baseline is to choose the right chain structure for each sentence

van Zaanen, 00	35.6	

Idea: Distributional Syntax?

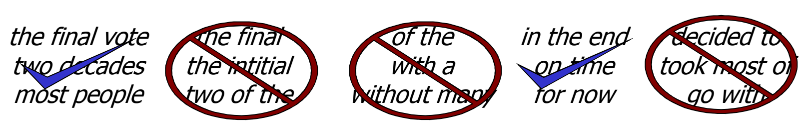
- Can we use distributional clustering for learning syntax? [Harris, 51]



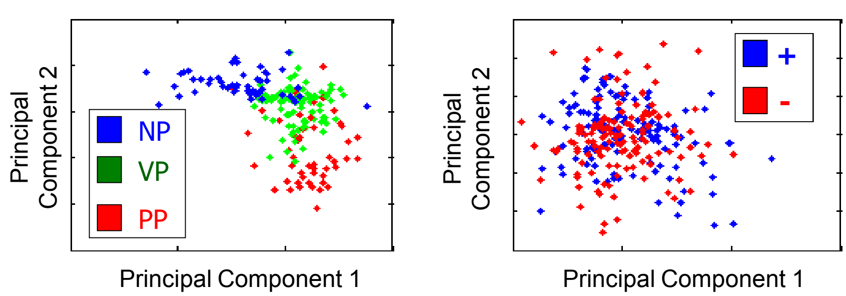
Span	Context
fell in september	*payrolls __ ♦*
payrolls fell in	*factory __ sept*

Problem: Identifying Constituents

Distributional classes are easy to find…

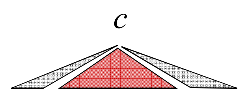


… but figuring out which are constituents is hard.

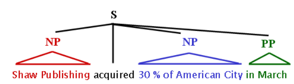


A Nested Distributional Model

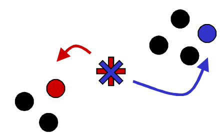
- We'd like a model that:
  - Ties spans to linear contexts (like distributional clustering)



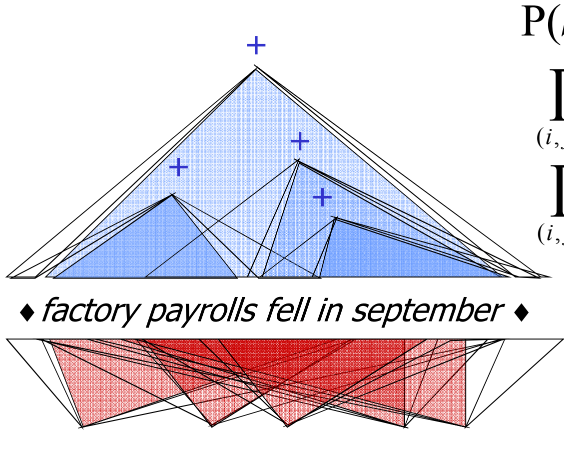
  - Considers only proper tree structures (like a PCFG model)



  - Has no symmetries to break (like a dependency model)



Constituent-Context Model (CCM)



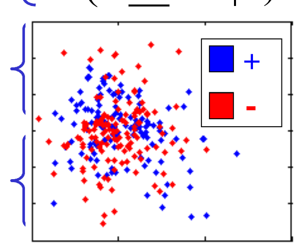
$$P(S|T) = \prod_{(i,j)\in T} P(\phi_{ij}\,|+)P(\chi_{ij}\,|+) \prod_{(i,j)\notin T} P(\phi_{ij}\,|-)P(\chi_{ij}\,|-)$$

P(*fpfis*|+)

P(♦__♦|+)

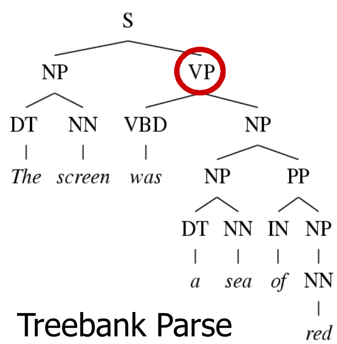
P(*fp*|+)

P(♦__ *fell*|+)

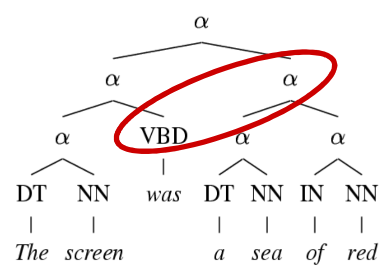


Results: Constituency

Right-Branch	70.0	



Treebank Parse



CCM Parse

Spectrum of Systematic Errors

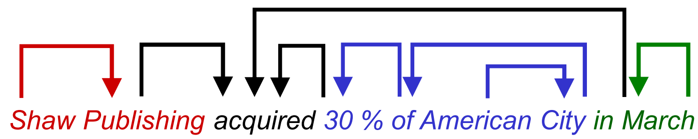
CCM analysis better ⟷ Treebank analysis better

Analysis	Inside NPs	Possesives	Verb groups
CCM	*the [lazy cat]*	*John ['s cat]*	*[will be] there*
Treebank	*the lazy cat*	*[John 's] cat*	*will [be there]*
CCM Right?	Yes	Maybe	No

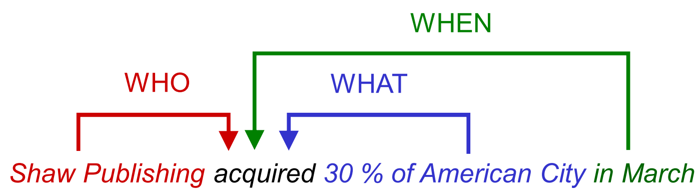
But the worst errors are the non-systematic ones (~25%)

Syntactic Parsing

- Parsing assigns structures to sentences.

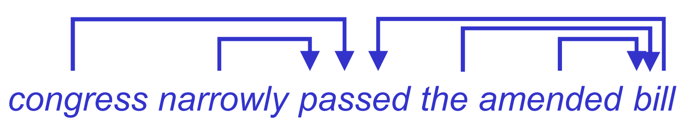


- Dependency structure gives attachments.



Idea: Lexical Affinity Models

- Words select other words on syntactic grounds



- Link up pairs with high mutual information
  - [Yuret, 1998]: Greedy linkage
  - [Paskin, 2001]: Iterative re-estimation with EM
- Evaluation: compare linked pairs to a gold standard

Method	Accuracy
Paskin, 2001	39.7

Problem: Non-Syntactic Affinity

- Mutual information between words does not necessarily indicate syntactic selection.

congress narrowly passed the amended bill

expect brushbacks but no beanballs

a new year begins in new york

Idea: Word Classes

- Individual words like congress are entwined with semantic facts about the world.
- Syntactic classes, like NOUN and ADVERB are bleached of word-specific semantics.
- Automatic word classes more likely to look like DAYS-OF-WEEK or PERSON-NAME.
- We could build dependency models over word classes. [*cf*. Carroll and Charniak, 1992]

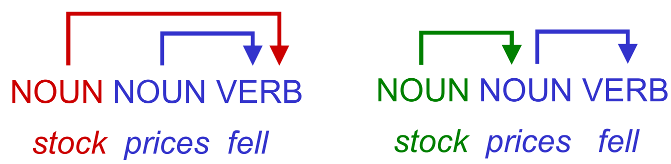
congress narrowly passed the amended bill

Problems: Word Class Models

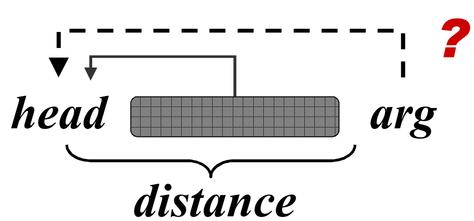
Random	41.7	
Carroll and Charniak, 92	44.7	

- Issues:
  - Too simple a model – doesn't work much better supervised
  - No representation of valence (number of arguments)

congress narrowly passed the amended bill



Local Representations



	Classes?	Distance	Local Factor
Paskin 01	✗	✗	$P(a \mid h)$

Common Errors: Dependency

Overproposed Dependencies

DET ← N	3474
N-PROP ← N-PROP	2096
NUM → NUM	760
PREP ← DET	735
DET ← N-PL	696
DET → PREP	627
DET → V-PAST	470
DET → V-PRES	420

Underproposed Dependencies

DET → N	3079
N-PROP → N-PROP	1898
PREP ← N	838
N → V-PRES	714
DET → N-PL	672
N ← PREP	669
NUM ← NUM	54
N → V-PAST	54

Results: Dependencies

Adjacent Words	55.9	
DMV	62.7	

- Situation so far:
  - Task: unstructured text in, word pairs out
  - Previous results were below baseline
  - We modeled word classes [*cf.* Carroll & Charniak 92]
  - We added a model of distance [*cf.* Collins 99]
  - Resulting model is substantially over baseline
  - … but we can do much better

Results: Combined Models

Dependency Evaluation (Undir. Dep. Acc.)

Random	45.6
DMV	62.7
CCM + DMV	64.7

Constituency Evaluation (Unlabeled Recall)

Random	39.4
CCM	81.0
CCM + DMV	88.0

- Supervised PCFG constituency recall is at 92.8
- Qualitative improvements
  - Subject-verb groups gone, modifier placement improved

How General is This?

Constituency Evaluation

English (7422 sentences)	
Random Baseline	39.4
CCM+DMV	88.0

German (2175 sentences)	
Random Baseline	49.6
CCM+DMV	89.7

Chinese (2473 sentences)	
Random Baseline	35.5
CCM+DMV	46.7

DMV	54.2
CCM+DMV	60.0

Dependency Evaluation