# Unsupervised Learning of Syntactic Structure

CS224N

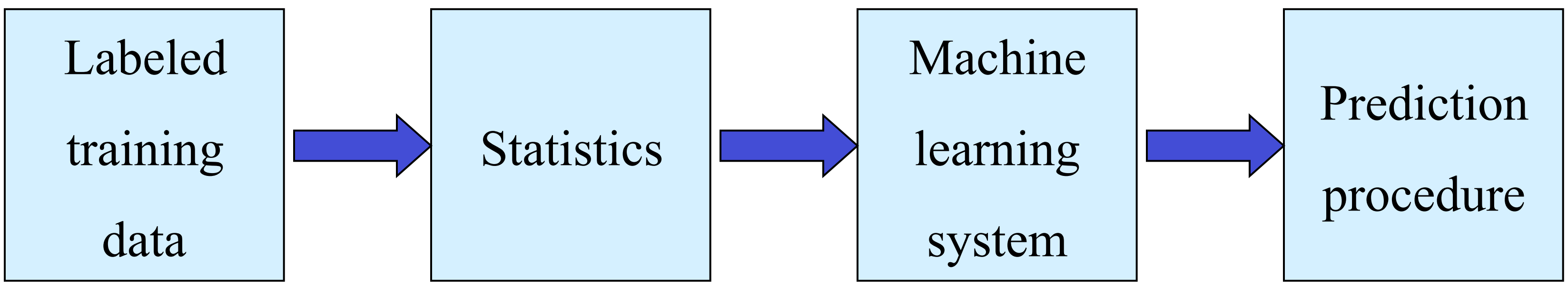
Christopher Manning

(borrowing slides from Dan Klein and Roger Levy)

# Supervised training

- Standard statistical systems use a supervised paradigm.

Training:



# The real story

- Annotating labeled data is *labor-intensive!!!*

Training:

Human effort → Labeled training data → Statistics → Machine learning system → Prediction procedure

# The real story (II)

- This also means that *moving to a new language, domain, or even genre can be difficult.*
- But unlabeled data is cheap!
- It would be nice to use the unlabeled data directly to learn the labelings you want in your model.
- Today we'll look at methods for doing exactly this for syntax learning.

# Learning structured models

- Most of the models we look at in this class have been *structured*
  - Tagging
  - NER
  - Parsing
  - Role labeling
- The structure is *latent*
- With raw data, we have to construct models that will *be rewarded for inferring that latent structure*

# A very simple example

- Suppose that we observe the following counts

| | |
|---|---|
| A | 9 |
| B | 9 |
| C | 1 |
| D | 1 |

- Suppose we are told that these counts arose from tossing two coins, each with a different label on each side
- Suppose further that we are told that the coins are not extremely unfair
- There is an intuitive solution; how can we learn it?

# A very simple example (II)

| | |
|---|---|
| A | 9 |
| B | 9 |
| C | 1 |
| D | 1 |

- Suppose we fully parameterize the model:

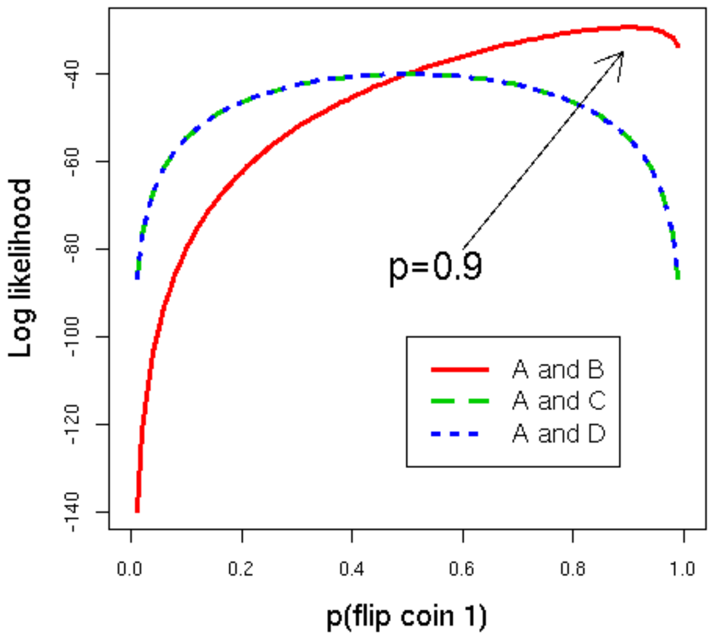
$$\pi \text{ :Probability of flipping coin 1}$$
$$p_1 \text{ :Probability of coin 1 coming up ``heads''}$$
$$p_2 \text{ :Probability of coin 2 coming up ``heads''}$$

- The MLE of this solution is totally degenerate: it cannot distinguish which letters should be paired on a coin
  - Convince yourself of this!
- We need to specify more constraints on the model
  - The general idea would be to place priors on the model parameters
  - An extreme variant: force $p_1=p_2=0.5$

# A very simple example (III)

| | |
|---|---|
| A | 9 |
| B | 9 |
| C | 1 |
| D | 1 |

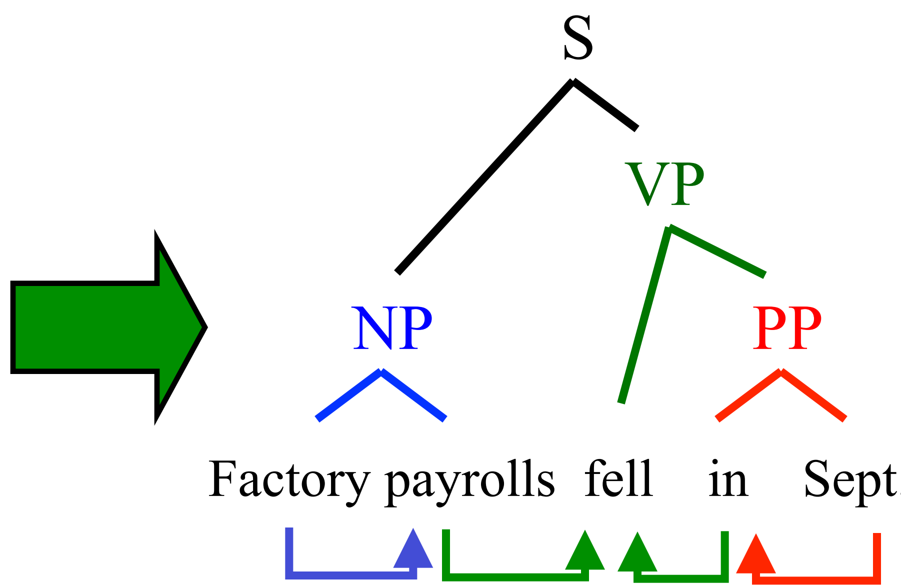
- An extreme variant: force $p_1=p_2=0.5$
- This *forces structure into the model*
- It also makes it easy to visualize the log-likelihood as a function of the remaining free para
- The intuitive solution is found!



# What is Grammar Induction?

- Given:
  - lots of (flat, linear) text of a language
  - but no knowledge particular to that language
- Goal: to discover the natural units of text (constituents)
- Parsing assigns structures to sentences and shows semantic modification relationships

```
He was previously vice president
Pick a country any country
Factory payrolls fell in September
South Korea has different concerns
The Artist has his routine
He is his own man
One claims he 's pro-choice
...
...
...
...
...
Who 's telling the truth
```



# Gold (1967)

- Gold: no superfinite class of languages (e.g., regular or context-free languages, etc.) is learnable without negative examples.
  - Certain conditions: nearly arbitrary sequence of examples; only constraint is that no sentence may be withheld from the learner indefinitely.
- Still regularly cited as bedrock for innatist linguistics
- Responses suggested by Gold:
  - Subtle, covert negative evidence ← Some recent claims
  - Innate knowledge shrinks language class ← Chomsky
  - Assumption about presentation of examples is too general ← e.g., probabilistic language model

# Horning (1969)

- If texts are generated by a stochastic process, how often something occurs can drive language acquisition
  - As time goes by without seeing something, we have evidence that it either doesn't happen or is very rare
  - Implicit negative evidence: can't withhold common stuff
- Horning: stochastic context-free languages are learnable* from only positive examples.
- But Horning's proof is enumerative, rather than providing a plausible grammar learning method

(See, e.g., Rohde and Plaut 1999 for discussion)

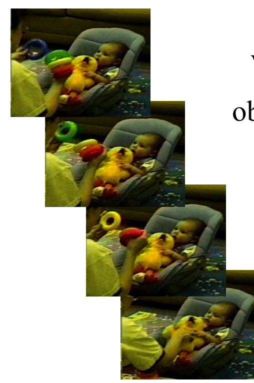
- Here we provide two case studies:
  1. Phrase structure (and dependencies) learning
  2. Learning semantic roles underlying surface syntax

# Motivations

- Natural Language Processing
  - Want to be able to make use of grammatical structure-based models for text domains and languages for which there are no treebanks (which is most of them)
  - There's much more data than annotated data
- Machine Learning
  - How can one learn structures, not just weights
- Linguistics/Cognitive Science
  - Unsupervised grammar learning can shed light on language acquisition and linguistic structure

# Is this the right problem?

- Before proceeding…are we tackling the right problem?
  - We’ve already seen how the structures giving high likelihood to raw text may not be the structures we want
  - Also, it’s fairly clear that kids don’t learn language structure from linguistic input alone…



words: “blue rings”

objects: rings, big bird

words: “and green rings”

objects: rings, big bird

words: “and yellow rings”

objects: rings, big bird

words: “Bigbird! Do you want to hold the rings?”

objects: big bird

  - Real learning is *cross-modal*
- So why study unsupervised grammar induction from raw text?

# Why induce grammars from raw text?

- I don’t think this is a trivial question to answer. But there *are* some answers:
  - Practical: if unsupervised grammar induction worked well, it’d save a whole lot of annotation effort
  - Practical: we need tree structure to get compositional semantics to work
  - Scientific: unsupervised grammar induction may help us a place an *upper bound* on how much grammatical knowledge must be innate
  - Theoretical: the models and algorithms that make unsupervised grammar induction work well may help us with other, related tasks (e.g., machine translation)

# Distributional clustering for syntax

- How do we evaluate unsupervised grammar induction?
- Basically like we evaluate supervised parsing: through bracketing accuracy
- But this means that we don't have to focus on inducing a *grammar*
- We can focus on inducing constituents in sentences instead

# Distributional clustering for syntax (II)

- What makes a good constituent?
- Old insight from Zellig Harris (Chomsky's teacher): *something can appear in a consistent set of contexts*

    Sarah saw a dog standing near a tree.

    Sarah saw a big dog standing near a tree.

    Sarah saw me standing near a tree.

    Sarah saw three posts standing near a tree.

- Therefore we could try to characterize the space of possible constituents by the contexts they appear in
- We could then do clustering in this space

# Distributional clustering for syntax (III)

- Clark, 2001: *k*-means clustering of common tag sequences occurring in context distributions
- Applied to 12 million words from the British National Corpus
- The results were promising:

(determiner)

"the big dog"

"the very big dog"

"the dog"

"the dog near the tree"

| | |
|---|---|
| AT0 AJ0 NN0 | AJ0 AJ0 |
| AT0 AJ0 NN1 | AJ0 CJC AJ0 |
| AT0 AJ0 NN2 | AV0 AJ0 |
| AT0 AV0 AJ0 NN1 | AV0 AV0 AJ0 |
| AT0 NN0 | ORD |
| AT0 NN1 PRP AT0 NN1 | |
| AT0 NN1 | |

# Problem: Identifying Constituents

Distributional classes are easy to find…

*the final vote*
*two decades*
*most people*

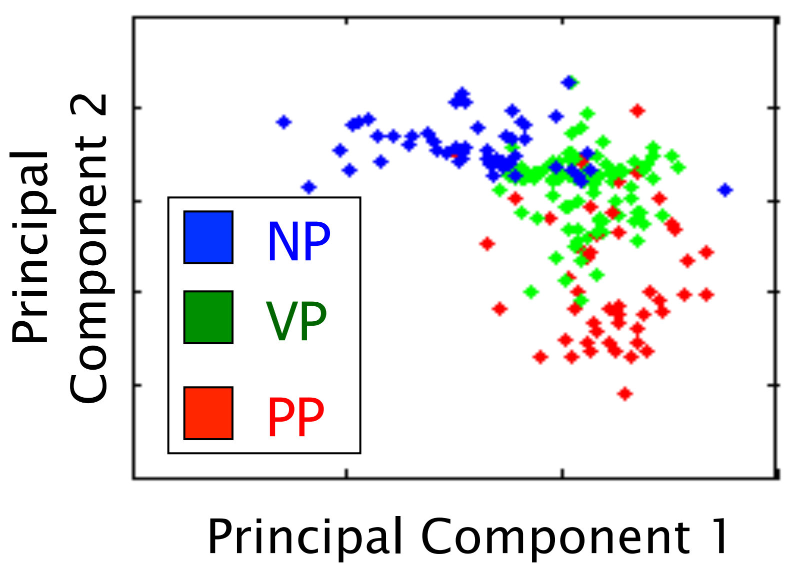
*the final*
*the intitial*
*two of the*

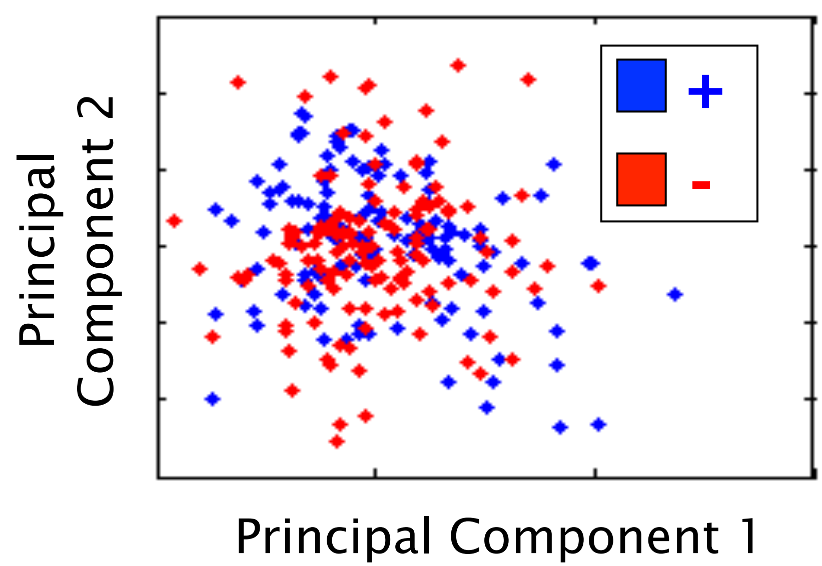
*of the*
*with a*
*without many*

*in the end*
*on time*
*for now*

*decided to*
*took most of*
*go with*

… but figuring out which are constituents is hard.



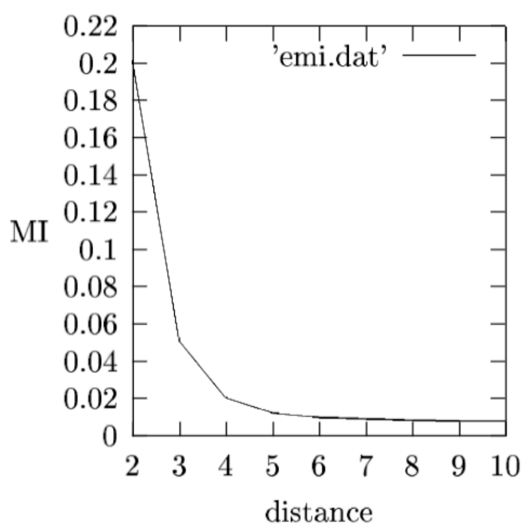



# Distributional clustering for syntax (V)

- Clark 2001's solution:

  "with real constituents, there is high mutual information between the symbol occurring *before* the putative constituent and the symbol *after*…"

  - Filter out distituents on the basis of low mutual information
  - (controlling for distance between symbols)




| | Cluster | Actual MI | Exp. MI | Valid |
|---|---|---|---|---|
| constituents | AT0 NN1 | 0.11 | 0.04 | Yes |
| | AT0 NP0 NP0 | 0.13 | 0.02 | Yes |
| | PRP AT0 NN1 | 0.06 | 0.02 | Yes |
| | AV0 AJ0 | 0.27 | 0.1 | Yes |
| distituents | NN1 AT0 | 0.008 | 0.02 | No |
| | AT0 AJ0 | 0.02 | 0.03 | No |
| | VBI AT0 | 0.01 | 0.02 | No |
| | PRP AT0 | 0.01 | 0.03 | No |

# Distributional clustering for syntax (VI)

- The final step: train an SCFG using vanilla maximum-likelihood estimation
- (hand-label the clusters to get interpretability)

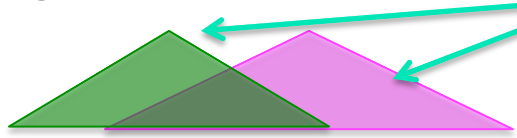
Ten most frequent rules expanding "NP"

| Count | Right Hand Side |
|---|---|
| 255793 | AT0 NN1 |
| 104314 | NP PP |
| 103727 | AT0 AJ0 NN1 |
| 73151 | AT0 NN2 |
| 72686 | DPS NN1 |
| 52202 | AJ0 NN2 |
| 51575 | DT0 NN1 |
| 35473 | NP NP |
| 34523 | DT0 NN2 |
| 34140 | AV0 NP |

# Structure vs. parameter search

- This did pretty well
- But the clustering-based structure-search approach is throwing out a hugely important type of information
- (did you see what it is?)
- …bracketing consistency!

*both can't be a constituent at once*

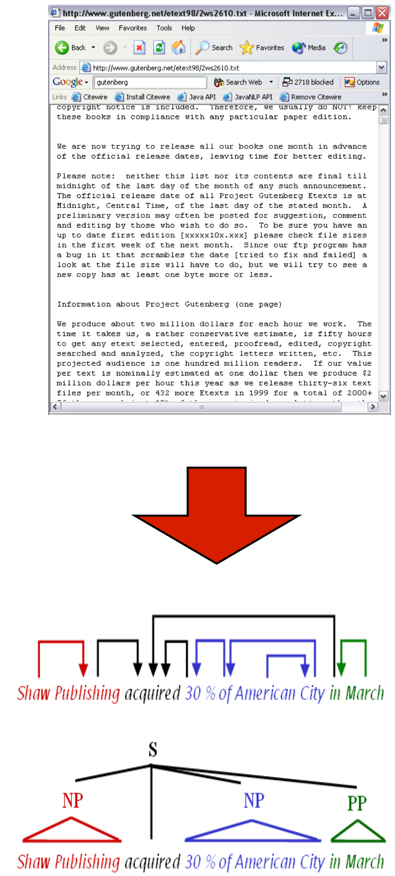


Sarah gave the big dog the most enthusiastic hug.

- Parameter-estimation approaches automatically incorporate this type of information
- This motivated the work of Klein & Manning (2002/2004)

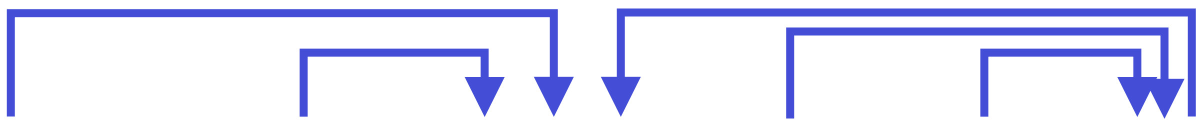
# Grammar induction: Klein and Manning (2004)

- 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
  - … but it *is* a hard problem



# Idea: Lexical Affinity Models

- Words select other words on syntactic grounds



*congress narrowly passed the amended bill*

- 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.

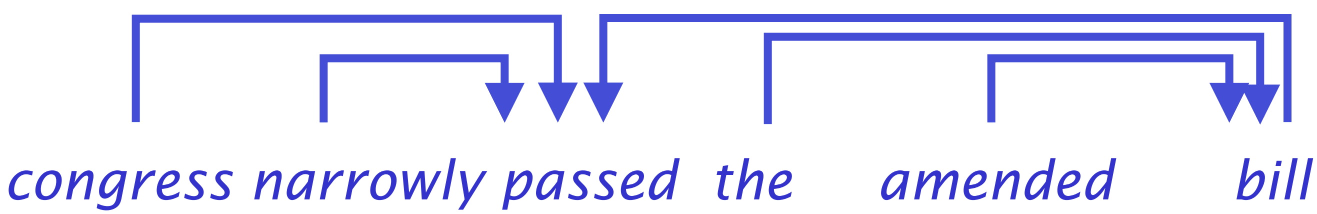
*congress narrowly passed the amended bill*

*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]

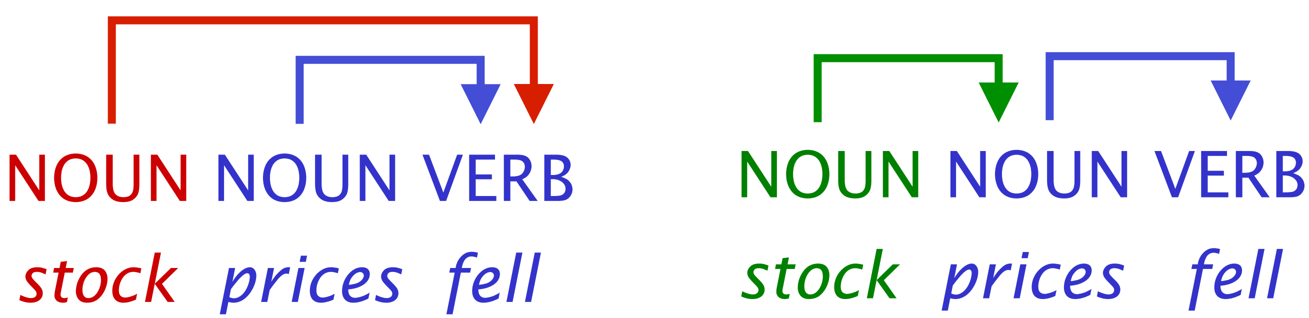


# Problems: Word Class Models

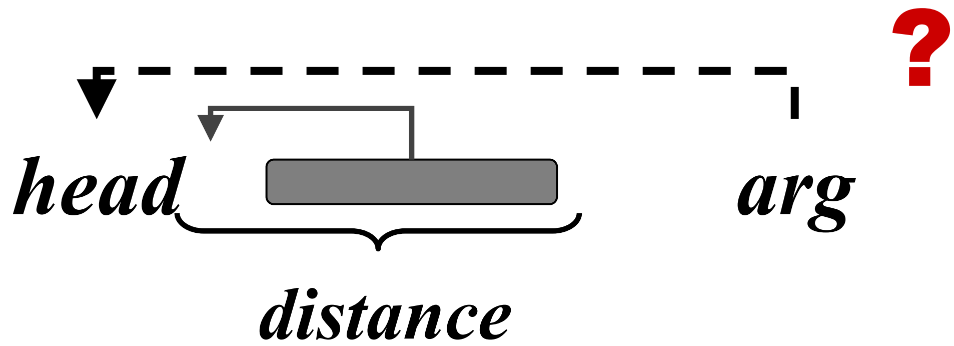
| | | |
|---|---|---|
| Random | 41.7 | |
| Carroll and Charniak, 92 | 44.7 | |

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

*congress narrowly passed the amended bill*



# Bias: Using more sophisticated dependency representations

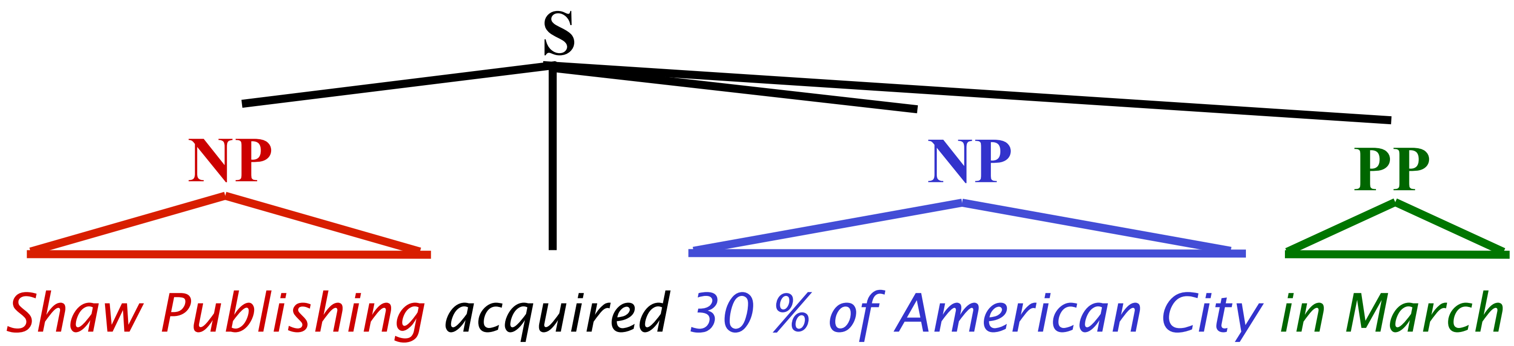



| | Classes? | Distance | Local Factor |
|---|---|---|---|
| Paskin 01 | ✗ | ✗ | $\mathrm{P}(a \mid h)$ |
| Carroll & Charniak 92 | ✓ | ✗ | $\mathrm{P}(c(a) \mid c(h))$ |
| Our Model (DMV) | ✓ | ✓ | $\mathrm{P}(c(a) \mid c(h), d)$ |

| | | |
|---|---|---|
| Adjacent Words | 55.9 | |
| Our Model (DMV) | 63.6 | |

# Constituency Parsing

- Model still doesn't directly use constituency




- Constituency structure gives boundaries

Information Extraction

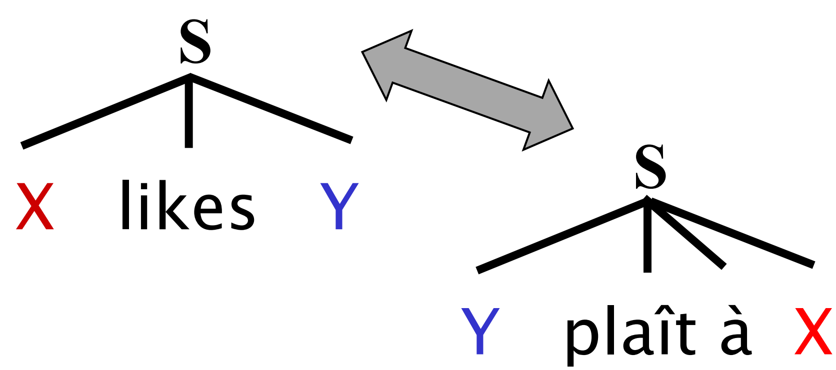
*30 % of American City*

~~*30 %*~~

~~*American City*~~

~~*30 % of American*~~

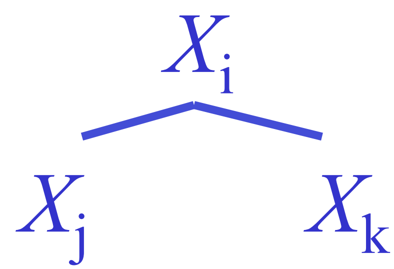
Machine Translation



# Idea: Learn PCFGs with EM

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

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

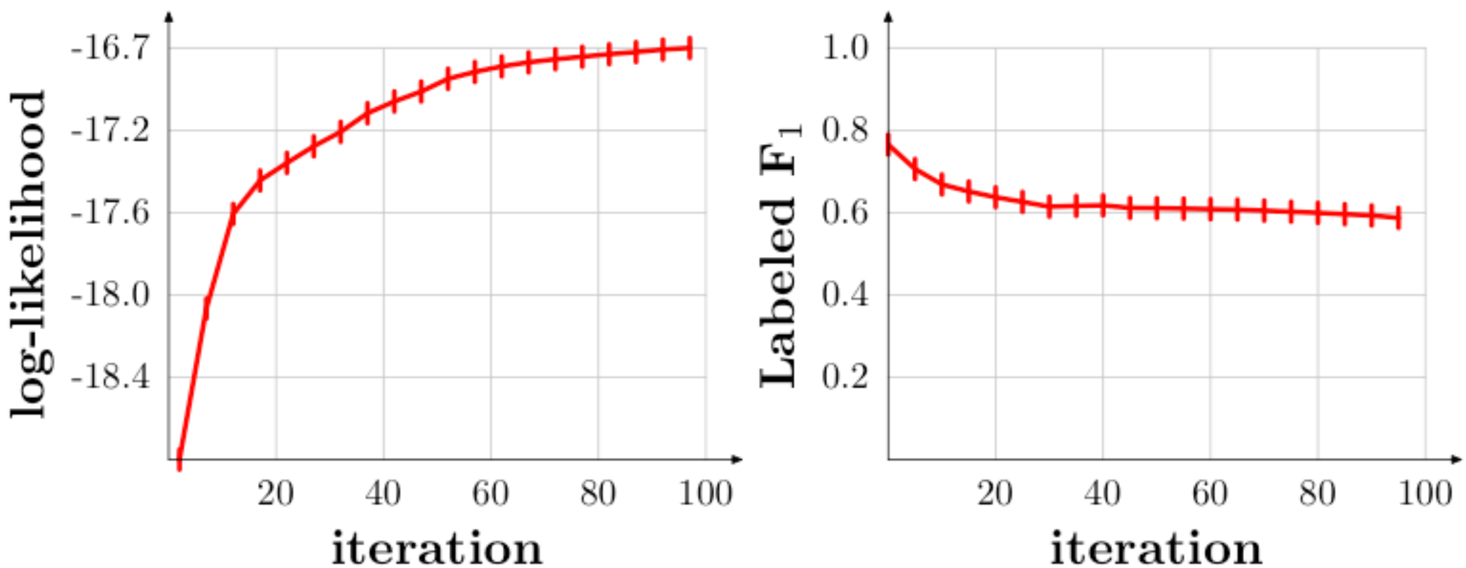



  - Full binary grammar over $n$ symbols
  - Parse randomly at first
  - Re-estimate rule probabilities off parses
  - Repeat
- Their conclusion: it doesn't work at all!

# Can we just use PCFGs?

- Early approaches: Pereira & Schabes (1992), Carroll & Charniak (1994)
  - Initialize a simple PCFG using some "good guesses"

PCFG (EM starting from supervised parameter estimate):

- 

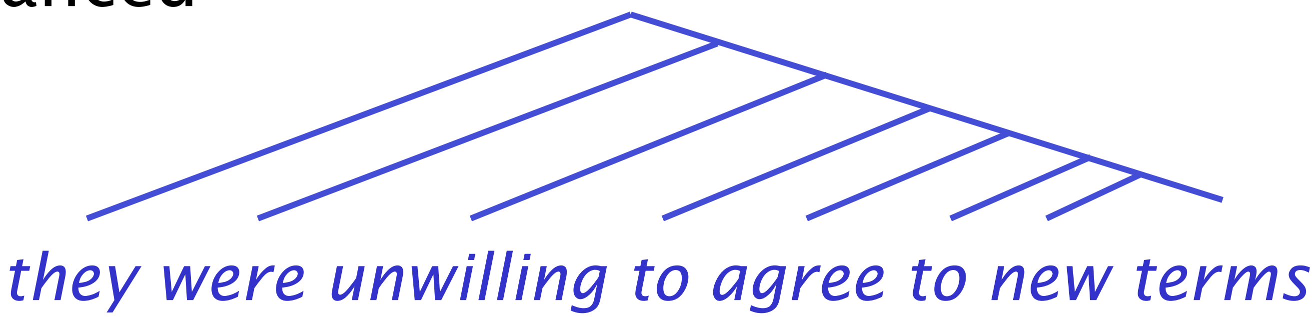
# Other Approaches

- Other earlier 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
  - GB/Minimalism No empirical results
- Evaluation: fraction of nodes in gold trees correctly posited in proposed trees (unlabeled recall)

| | | |
|---|---|---|
| Adriaans, 1999 | 16.8 | |
| Clark, 2001 | 34.6 | |
| van Zaanen, 2000 | 35.6 | |

# Right-Branching Baseline

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




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

| | | |
|---|---|---|
| van Zaanen, 00 | 35.6 | |
| Right-Branch | 46.4 | |

# Inspiration: 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*

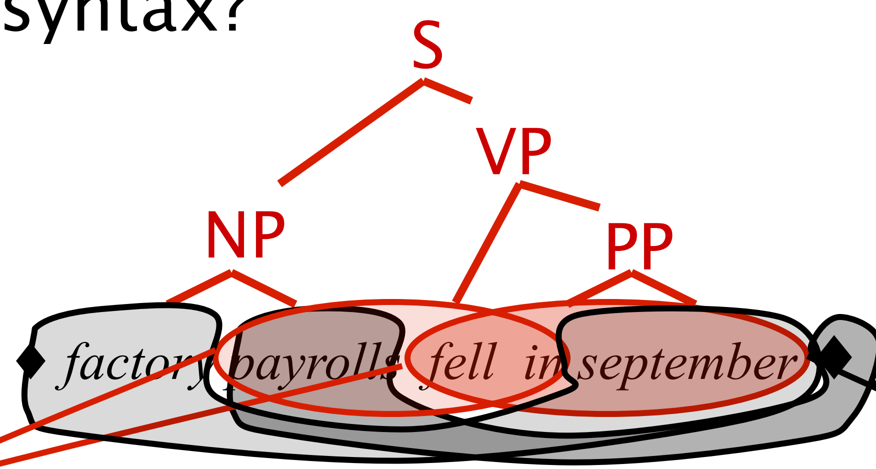
*the*
*a*

*said*
*reported*

[Finch and Chater 92, Schütze 93, Clark 01, many others]

# Idea: Distributional Syntax?

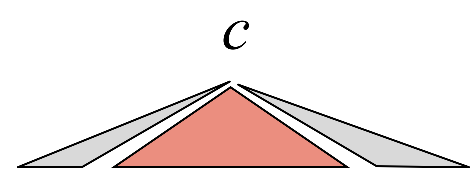
- Can we use distributional clustering for learning syntax?



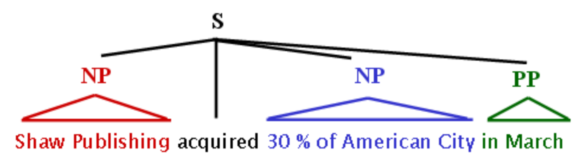

| Span | Context |
|---|---|
| *fell in september* | *payrolls __ ♦* |
| *payrolls fell in* | *factory __ sept* |

# A Nested Distributional Model

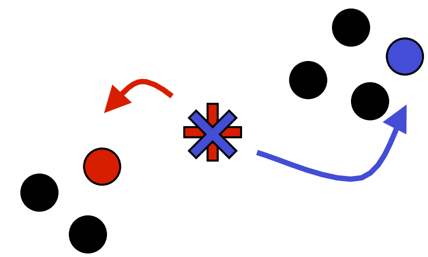
- We'd like a model that:
  - Ties spans to linear contexts (like distributional clustering)



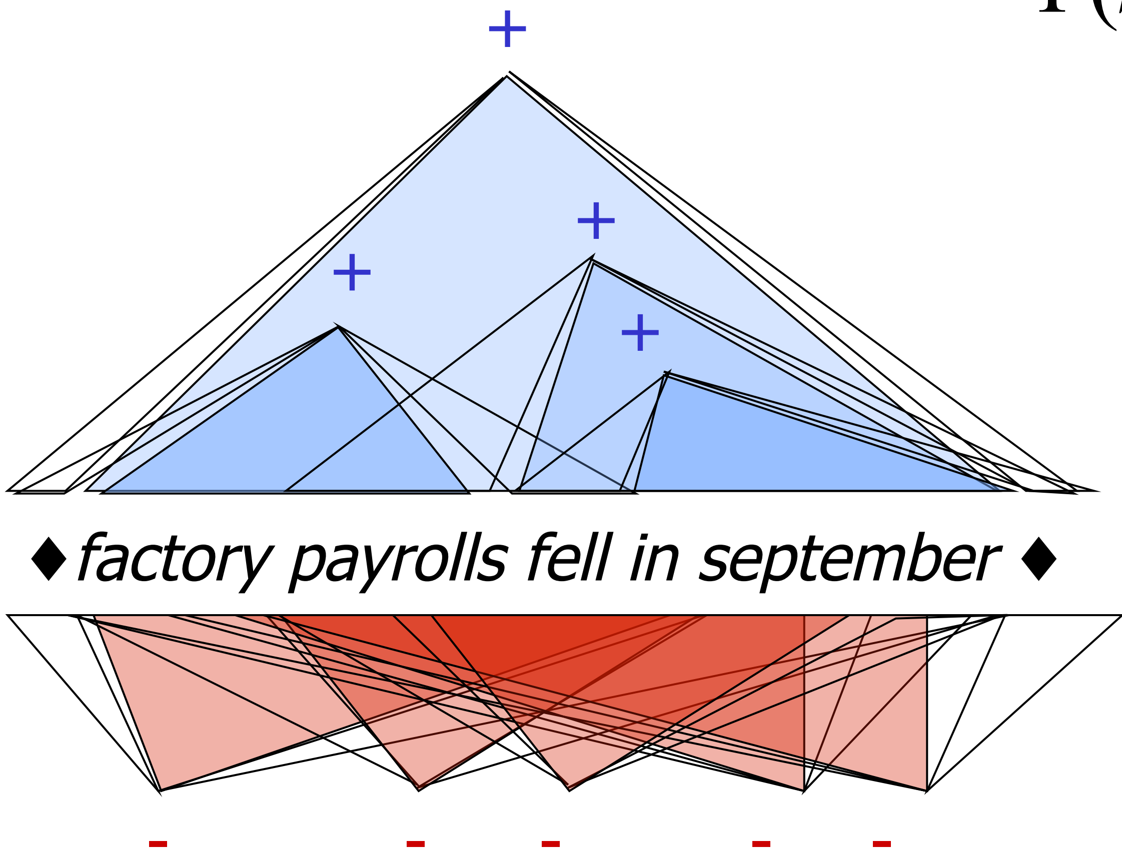

  - Considers only proper tree structures (like a PCFG model)




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



# Constituent-Context Model (CCM)




$\mathrm{P}(S|T) =$

$\mathrm{P}(\mathit{fpfis}|+)$
$\mathrm{P}(\blacklozenge\_\_\blacklozenge|+)$

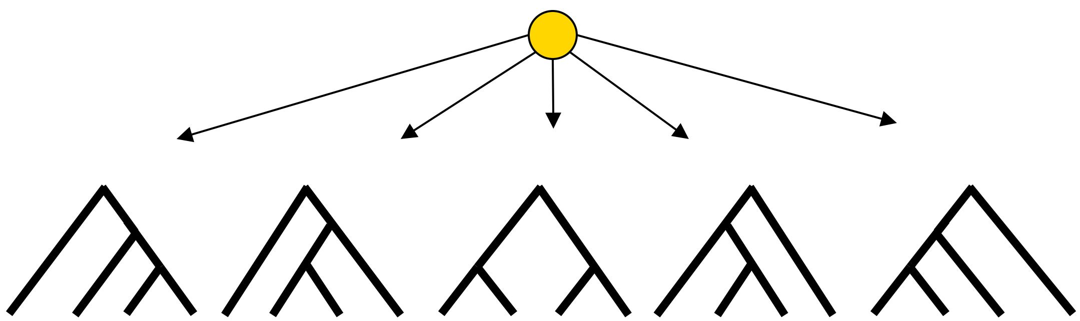
$\mathrm{P}(\mathit{fp}|+)$
$\mathrm{P}(\blacklozenge\_\_\ \mathit{fell}|+)$

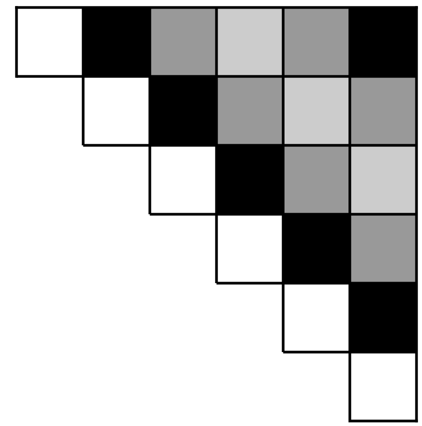
$\mathrm{P}(\mathit{fis}|+)$
$\mathrm{P}(\mathit{p}\ \_\_\ \blacklozenge|+)$

$\mathrm{P}(\mathit{is}|+)$
$\mathrm{P}(\mathit{fell}\ \_\_\ \blacklozenge|+)$

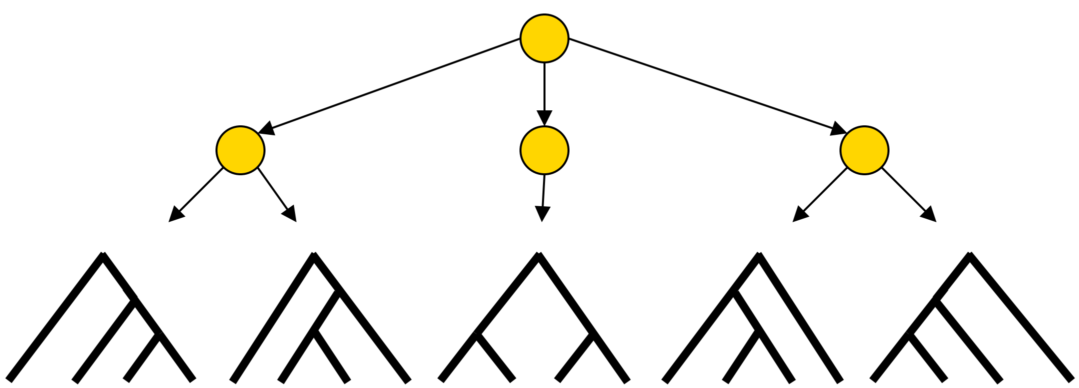
# Initialization: A little UG?

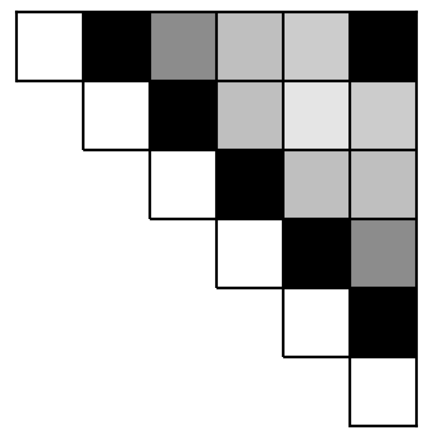
## Tree Uniform





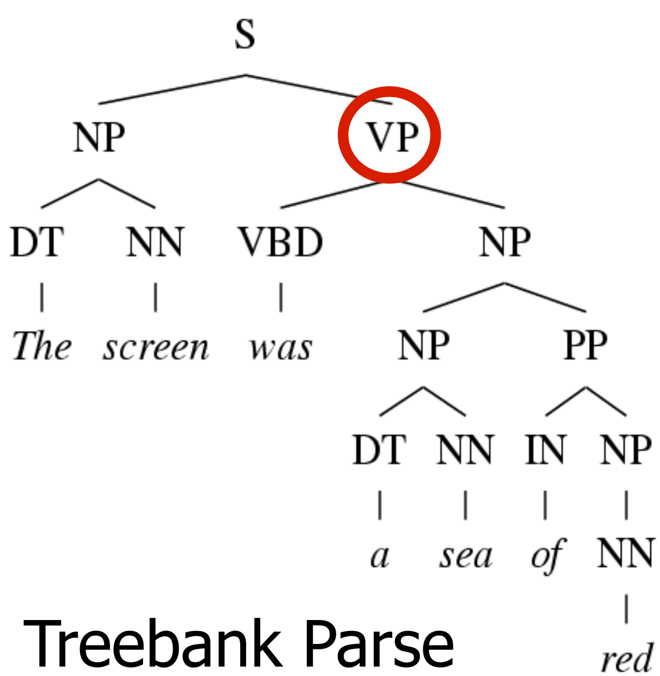
## Split Uniform



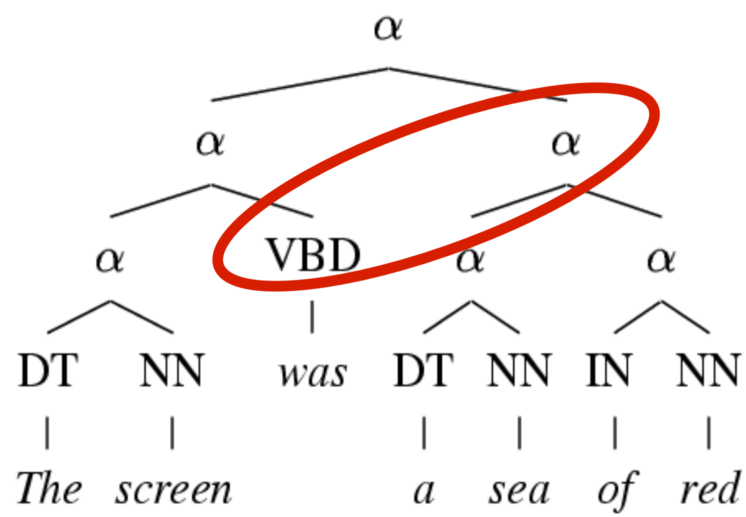


# Results: Constituency

| | | |
|---|---|---|
| Right-Branch | 70.0 | |



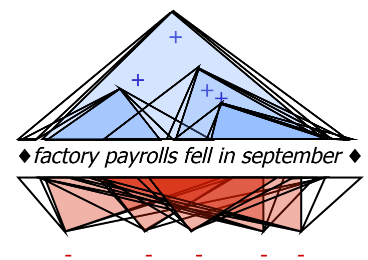

Treebank Parse

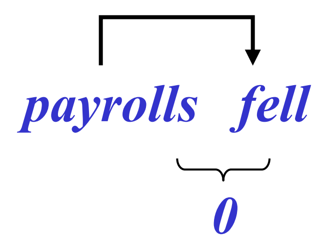



CCM Parse

# A combination model

- What we've got:
  - Two models of syntactic structure
  - Each was the first to break its baseline
- Can we combine them?
- Yes, using a product model
  - Which we also used for supervised parsing (Klein and Manning 2003)



# Combining the two models

[Klein and Manning ACL 2004]

## Dependency Evaluation

| | | |
|---|---|---|
| Random | 45.6 | |
| DMV | 62.7 | |
| CCM + DMV | 64.7 | |

## Constituency Evaluation

| | | |
|---|---|---|
| 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

# Crosslinguistic applicability of the learning algorithm

**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 |

# Most Common Errors: English

Overproposed Constituents

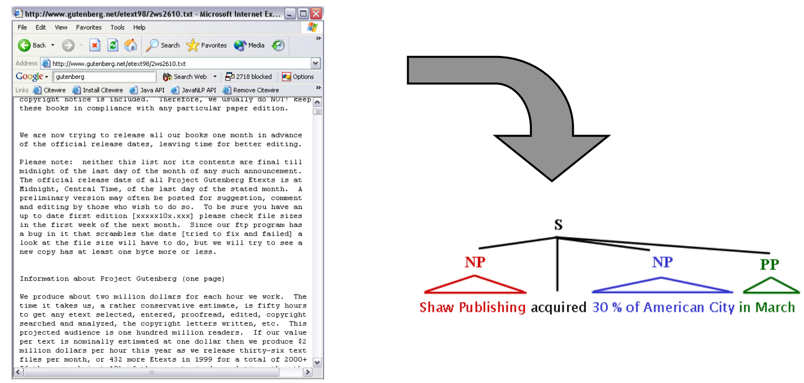
| | | |
|---|---|---|
| ADJ N | 1022 | *the [general partner]* |
| N-PROP N-PROP | 447 | *the [Big Board]* |
| DET N | 398 | *[an import] order* |
| ADJ N-PL | 294 | *six million [common shares]* |
| N-PL ADV | 164 | *[seats currently] are quoted* |

Crossing Constituents

| | | |
|---|---|---|
| NUM NUM PREP NUM NUM | 154 | *rose to [# billion from # billion]* |
| N-PL ADV | 133 | *petroleum [prices also] surged* |
| N-PROP N-PROP N-PROP | 67 | *to [Hong Kong China] is* |
| ADJ N | 66 | *especially [strong growth]* |

# What Was Accomplished?

- Unsupervised learning:
  - Constituency structure
  - Dependency structure




- Constituency recall:

| | | |
|---|---|---|
| Random Baseline | 39.4 | |
| CCM + DMV | 88.0 | |
| Supervised PCFG | 92.8 | |

- Why it works:
  - Combination of simple models
  - Representations designed for unsupervised learning

# More recently…

- Quite a bit of other work has built on these results
  - Smith and Eisner 2005
  - Headden et al. 2009
  - Cohen and Smith 2009
  - Spitkovsky et al. 2010
- Improved performance (on longer sentences!) via a variety of techniques.