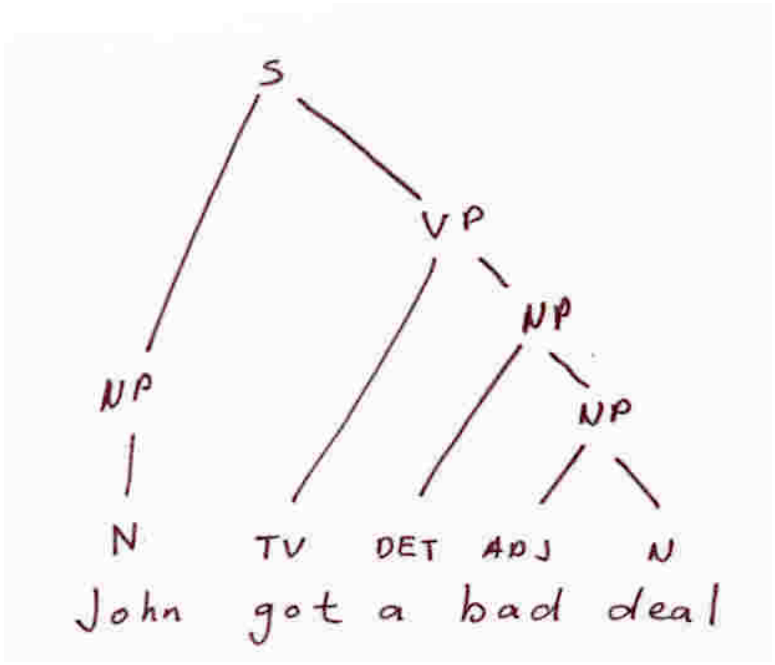
Standard and Non-Standard Parse Trees Equally Improve Grammar Induction
ISCOL 2008

Jonathan Berant
Ben Sandbank
Eytan Ruppin
Shimon Edelman

Grammar vs. Parser Induction

Parser Induction

John got a bad deal

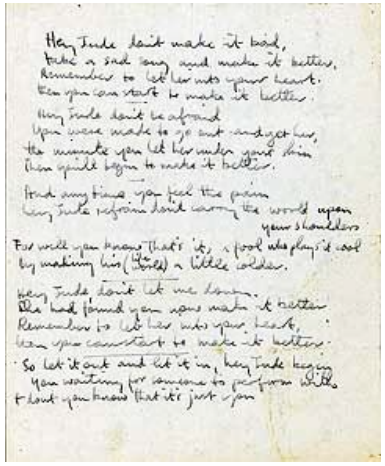


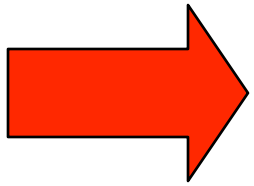
Motivation: mainly applicative

Measure: compare to some structural gold standard

Grammar vs. Parser Induction

Grammar Induction



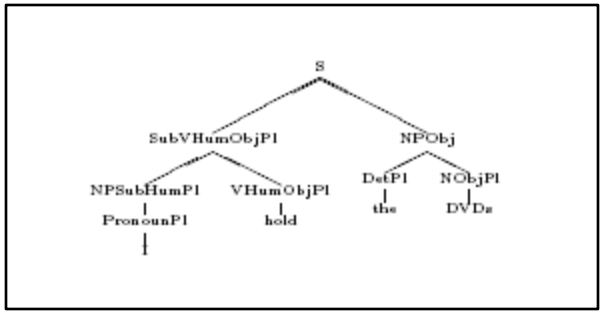


Motivation: mainly cognitive

Measure: generate grammatical sentences and discriminate between grammatical and ungrammatical sentences

The two are distinct!

I hold the dvds



a) Which problem will bill solve? (Bikel parser log probability: -56.858)

b) Bill solve which will problem? (Bikel parser log probability -53.267)

(Fong and Berwick, 2008; Okanohara and Tsujii, 2007)

University of California
C A L
N L P
Berkeley

Berkeley parser trained on an artificial grammar:

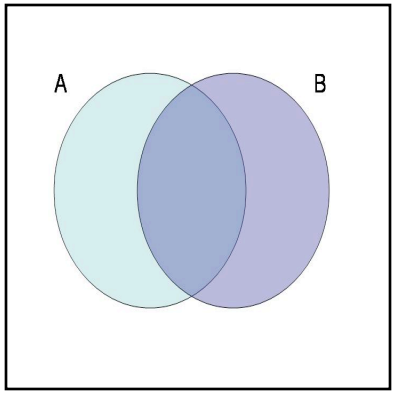
- High parsing scores (0.83 F-measure, an underestimate)
- Low grammar induction scores (0.11 F-measure)

Evaluation of unsupervised Grammar Induction

Recall: proportion of sentences sampled from the real language that are given non-zero probability by the learner.

Precision: proportion of sentences sampled from the generated language that are grammatical in the real language. (Solan et al., 2005).

***Problem*:** An adversary spreading *1- ε* probability over the train set and the rest of the probability mass over all sentences will get a perfect score.



Evaluation of unsupervised Grammar Induction

Perplexity: The average surprise of the model when encountering the sentences of the test set.

Problem: The model must be smoothed and assign non-zero probability to any sentence, otherwise it will be infinitely surprised when encountering an unlearned sentence. Evaluation is too sensitive to the type of smoothing.

To overcome evaluation difficulties we used an ensemble of measures

***Goal**: Explore the role of constituency in grammar induction*

***Method:** Use a general grammar induction algorithm to assess the effect of constituency information*

Previous work

- Structural knowledge use has been explored in the field of language modeling (Charniak, 2001; Roark, 2001; Xu et al., 2002)
- We don't use constituent labels only constituent structure
- Previous work used the perplexity measure for evaluation which is problematic in our setting.

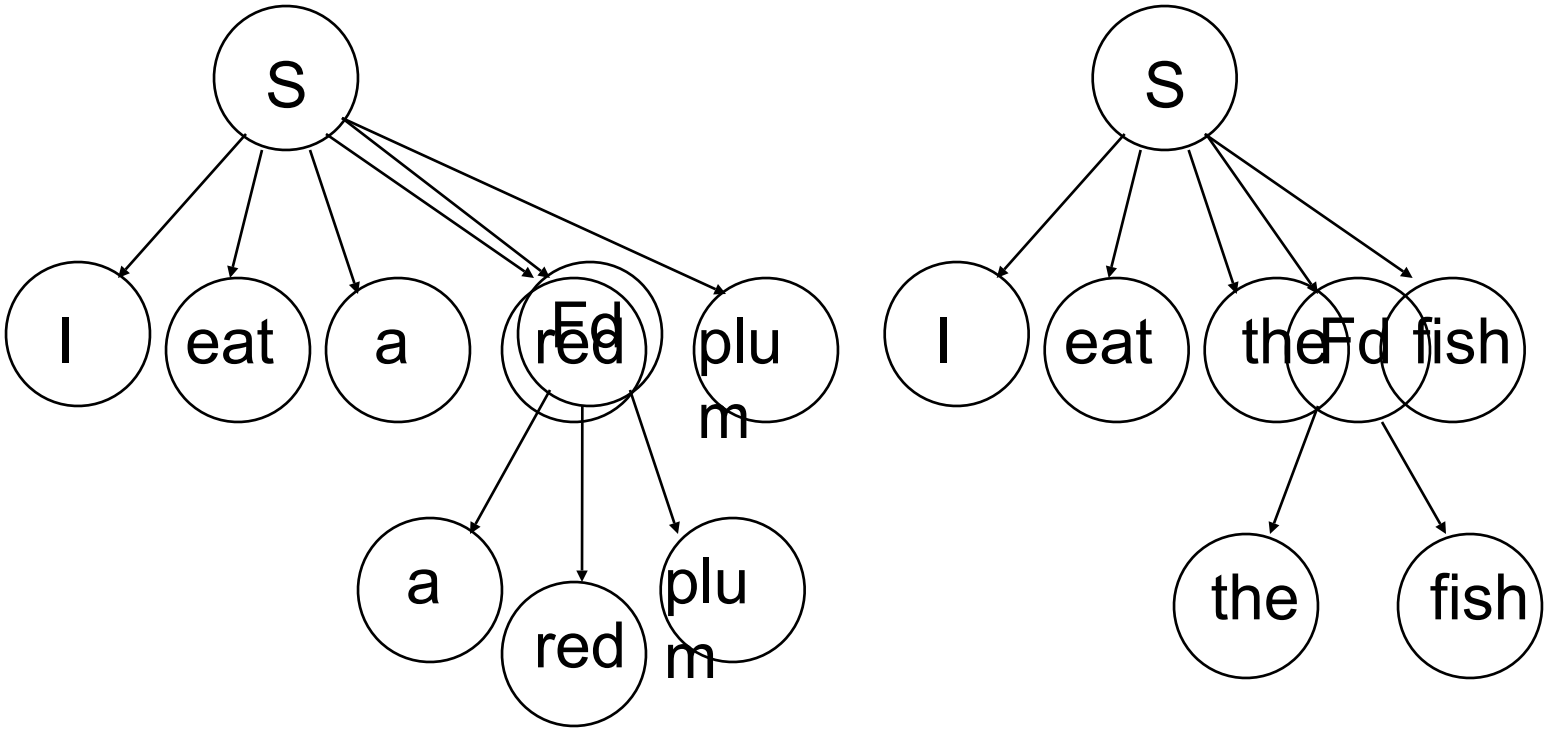
Corpora

- 10,000 sentences were generated from a few artificial grammars. Ten learners are trained on ten train/test set splits (n-fold testing).

1. [$_{S}$ [$_{NPSubHumSing}$ [$_{PN}$ Ben]] [$_{VPHumHumSing}$ hears] [$_{NPHumSing}$ [$_{NPHumSingSimp}$ [$_{DetSing}$ the] [$_{NHumSing}$ student]] [$_{CPHumSingGap}$ [$_{CHum}$ that] [$_{VPHumSing}$ [$_{VHumSing}$ runs]]]]]]
2. After they exercise, the men bend, although Joe carresses a director who the TVs that expand calm

Algorithm - ConText (Sandbank et al.)

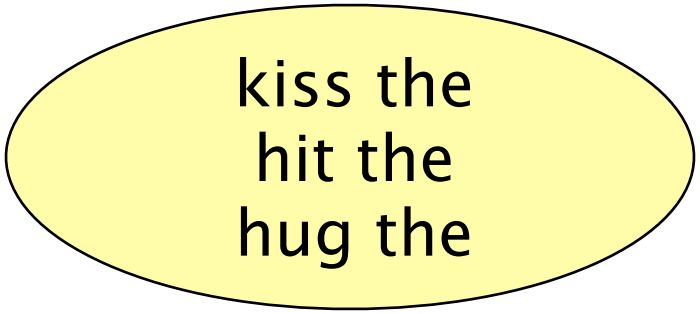
- A simple, general, distributional-clustering-based PCFG induction algorithm.



Baseline – raw ConText

Scenario	Recall	Precision	F_g	Perplexity	Parse recall	Parse precision	F_p
comp-shallow	0.94	0.63	0.76	23.39	0.45	0.53	0.48
comp-deep	0.93	0.62	0.74	23.54	0.34	0.55	0.42

C1:

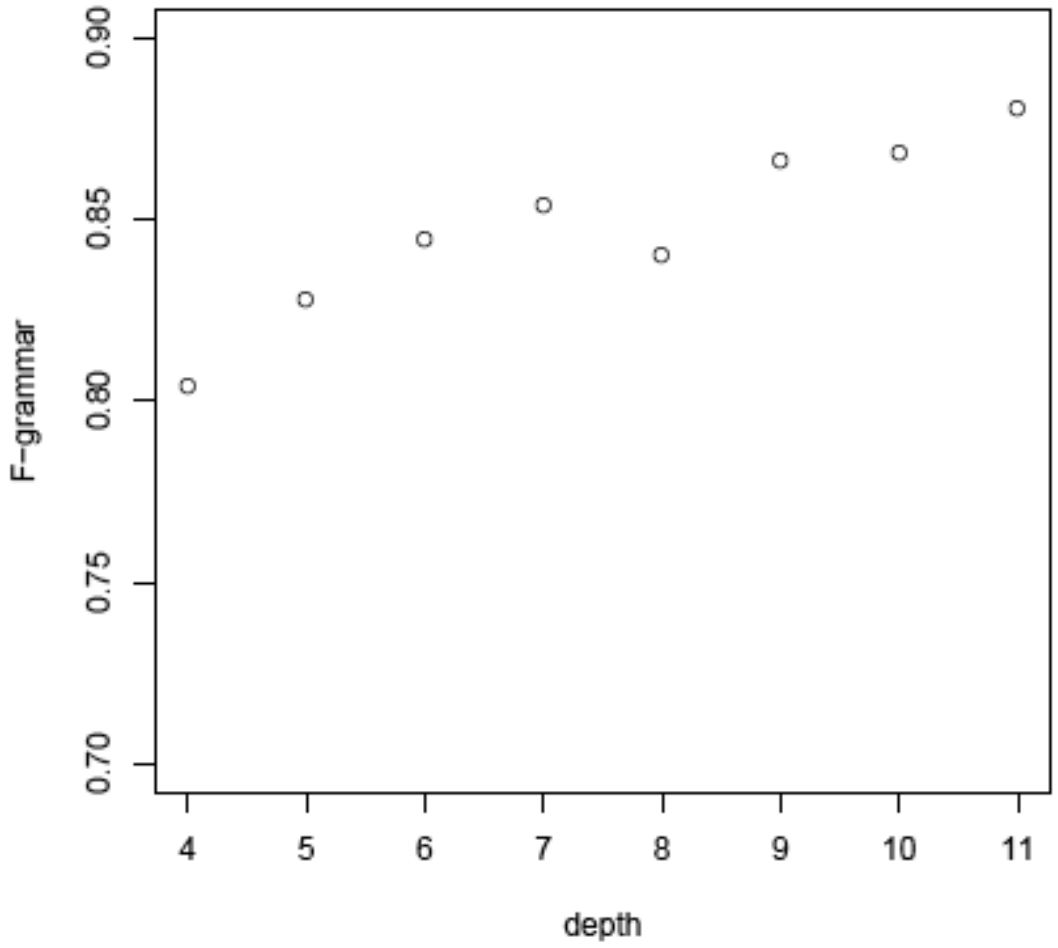


ConText* - Using constituency cues

- Consider only subsequences that do not violate the "true" constituency structure as candidates for clustering

Scenario	Recall	Precision	F_g	Perplexity	Parse recall	Parse precision	F_p
comp-shallow	0.94	0.63	0.76	23.39	0.45	0.53	0.48
comp-deep	0.93	0.62	0.74	23.54	0.34	0.55	0.42
comp-shallow*	0.9	0.8	0.85	24.45	0.61	0.76	0.68
Comp-deep*	0.88	0.87	0.88	26.13	0.49	1	0.66

Gradual Constituent Cues



Imprecise constituency cues

- We implemented the unsupervised CCM parser (Klein, 2005) and parsed 10,000 sentences (<10 words) from the *comp* grammar.

Scenario	Recall	Precision	F_g
comp-Klein	0.97	0.78	0.86
Comp-Klein*	0.93	0.91	0.92 P<(0.001)

Some hand-waving

- In the *comp* corpus "hug the", "kiss the" and "hit the" ARE substitutable.

"the man hugs the man"

S→NP VP (1.0)

NP → DET N (1.0)

VP → V NP (1.0)

Det → the (1.0)

N → man (1.0)

V → hugs (1.0)

S→NP VP (1.0)

NP → Det N (1.0)

VP → VD N (1.0)

Vd → V Det

Det → the (1.0)

N → man (1.0)

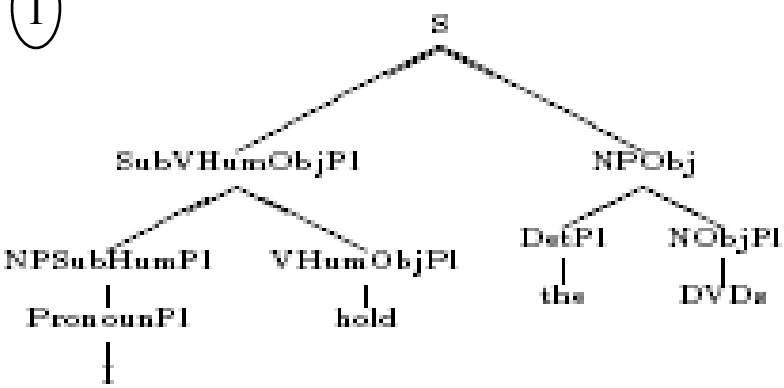
V → hugs (1.0)

Non-standard parse trees

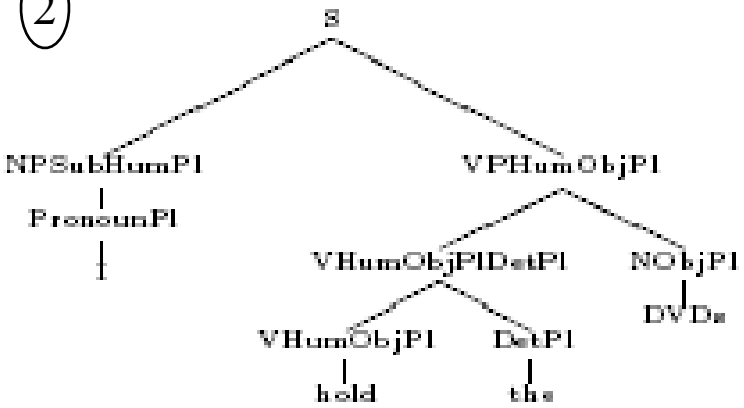
- Three types of constituent violations are common:
  - Subject-verb and not verb-object
  - Complementizers *(that,who,which)* attaching to the NP on the left and not to the subordinate clause
  - Determiners *(the,a)* attaching to the verb on the left and not the noun on the right

Non-standard parse trees

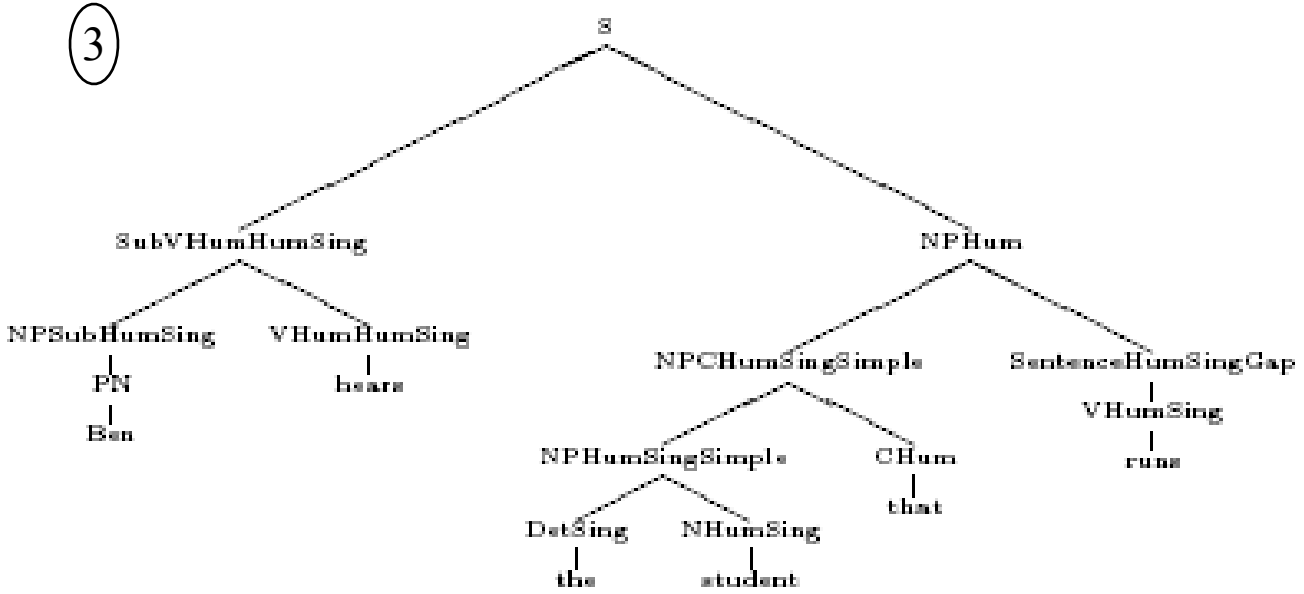
1



2



3



Non-standard parse trees

- All grammars generate the exact same language
- Re-parsing the 10,000 sentences with these grammars caused up to 24,000 constituent changes

Scenario	Recall	Precision	F_g	Perplexity	Parse recall	Parse precision	F_p
comp-deep	0.93	0.62	0.74	23.54	0.34	0.55	0.42
comp-sv*	0.9	0.84	0.87	24.53	0.54	1	0.7
comp-comp*	0.92	0.89	0.9	23.24	0.56	1	0.72
comp-det*	0.9	0.84	0.87	25.04	0.42	1	0.59

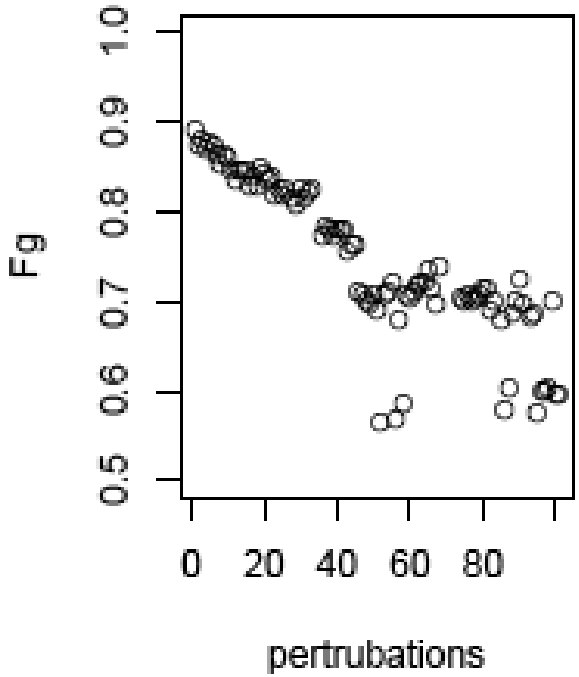
Random non-standard parse trees

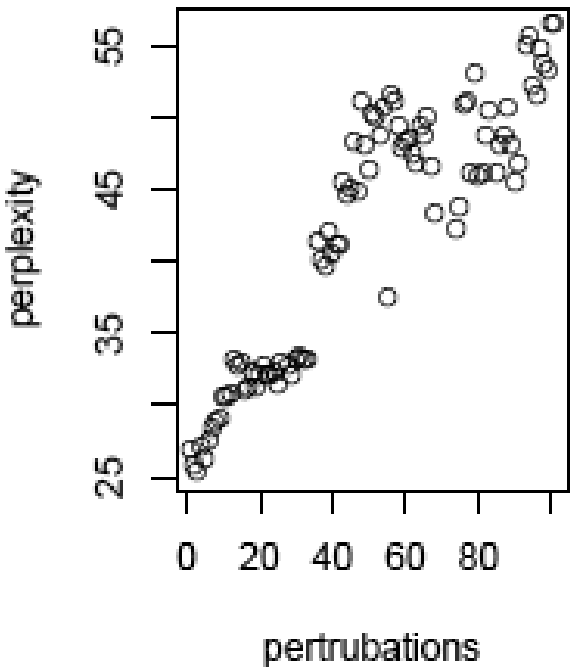
Procedure:

1. Generate a set D of all subsequences of pre-terminals that are never constituents and occur more than K times (K=100) in the comp treebank
2. Do 100 times:
    1. Remove a random pre-terminal subsequence S from D
    2. For every tree T that contains S: modify T in a minimal manner so that S is a constituent

Output: 100 treebanks that gradually diverge from the original treebank in a consistent manner

Random non-standard parse trees





Conclusion

- Constituency cues help grammar induction by constraining the subsequence search space
- Non-standard parse trees lead to good grammar induction performance (similar to Bod, 2007)
- Constituents might be overlapping which is not possible in CFGs.