Upper Bounds for Unsupervised Parsing with Unambiguous Non-Terminally Separated Grammars

Franco M. Luque Gabriel Infante-Lopez

Grupo de Procesamiento de Lenguaje Natural
Fa.M.A.F., Universidad Nacional de Córdoba

CONICET

CLAGI, 12th Conference of the EACL
March 30, 2009

Outline of the Talk

1. Introduction
2. UNTS Grammars
3. Finding the Best Grammar
   - The W Measure
4. Bounding the F1
5. Bounds for the WSJ10
6. Conclusions

Introduction

Design of Unsupervised Parsers

- A main issue is choosing a model.
- Models have learnability (L) and expresiveness (E) properties.
- Trade-off between L and E: Simple models are easier to learn than expressive models.

Introduction

Design of Unsupervised Parsers

- A main issue is choosing a model.
- Models have learnability (L) and expresiveness (E) properties.
- Trade-off between L and E: Simple models are easier to learn than expressive models.

UNTS Grammars (Clark, 2006 & 2007)

- L: Good properties (PAC-learnable in polynomial time).
- E: Natural Language is not strictly UNTS but is close to it.

Introduction

Design of Unsupervised Parsers

- A main issue is choosing a model.
- Models have learnability (L) and expresiveness (E) properties.
- Trade-off between L and E: Simple models are easier to learn than expressive models.

UNTS Grammars (Clark, 2006 & 2007)

- L: Good properties (PAC-learnable in polynomial time).
- E: Natural Language is not strictly UNTS but is close to it.

To what extent is UNTS close to Natural Language?

Introduction: Approach

To what extent is UNTS close to Natural Language?

Introduction: Approach

To what extent is UNTS close to Natural Language?

Measuring Closeness

- Use natural language *gold treebanks* as reference.
- Use standard measure for unsupervised parsing: Unlabeled F1. Results comparable with state-of-the art parsers.
- Consider all the ways UNTS grammars can parse the sentences.
- Find F1 upper bounds for the entire class of UNTS grammars. No hill-climbing. Hard upper bound independent of the learning algorithm.

UNTS Grammars: Definition

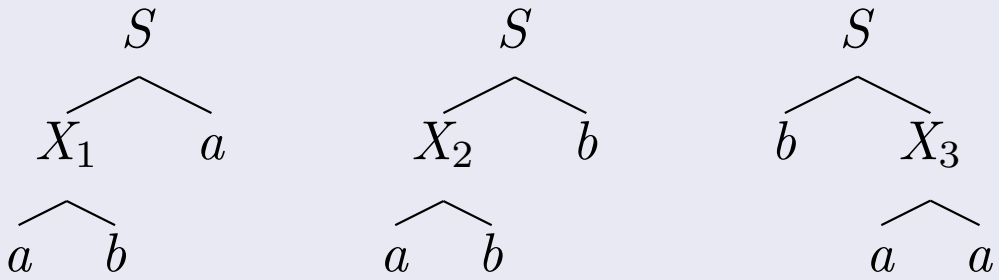
- Subclass of Deterministic CFGs.

UNTS Grammars: Definition

- Subclass of Deterministic CFGs.
- A substring is always a *constituent* or it is always not (it is a *distituent*).

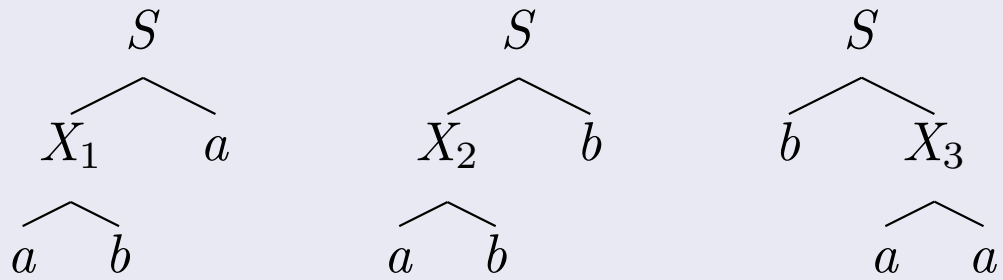
UNTS Grammars: Definition

- Subclass of Deterministic CFGs.
- A substring is always a *constituent* or it is always not (it is a *distituent*).
  - For instance: $L = \{aba, abb, baa\}$



UNTS Grammars: Definition

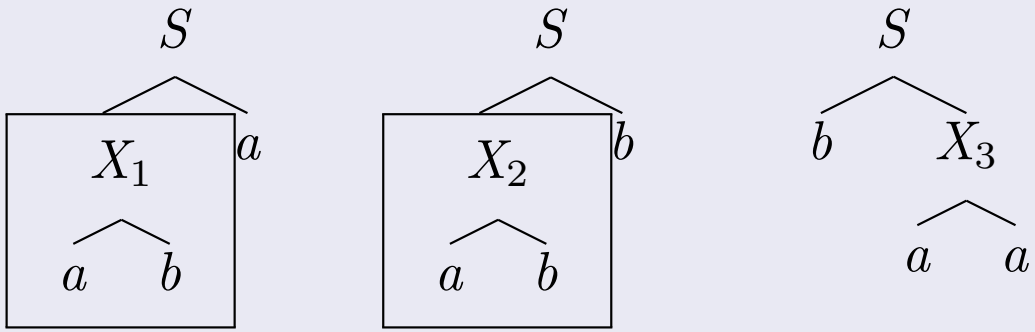
- Subclass of Deterministic CFGs.
- A substring is always a *constituent* or it is always not (it is a *distituent*).
  - For instance: $L = \{aba, abb, baa\}$



  - The constituents are $C = \{ab, aa, aba, abb, baa\}$.

UNTS Grammars: Definition

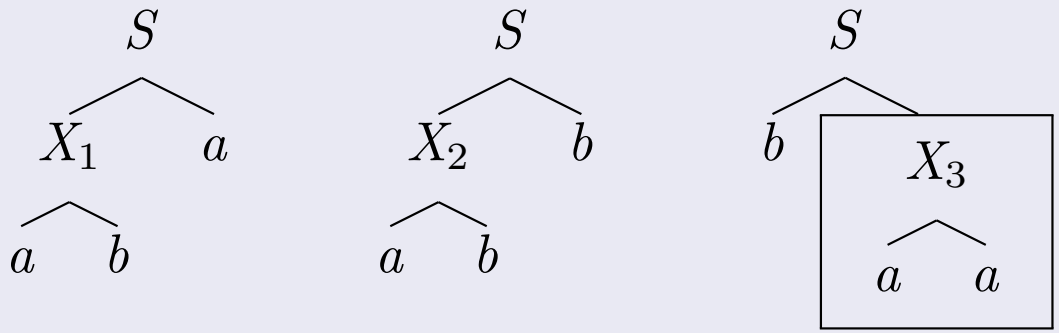
- Subclass of Deterministic CFGs.
- A substring is always a *constituent* or it is always not (it is a *distituent*).
  - For instance: $L = \{aba, abb, baa\}$



  - The constituents are $C = \{ab, aa, aba, abb, baa\}$.

UNTS Grammars: Definition

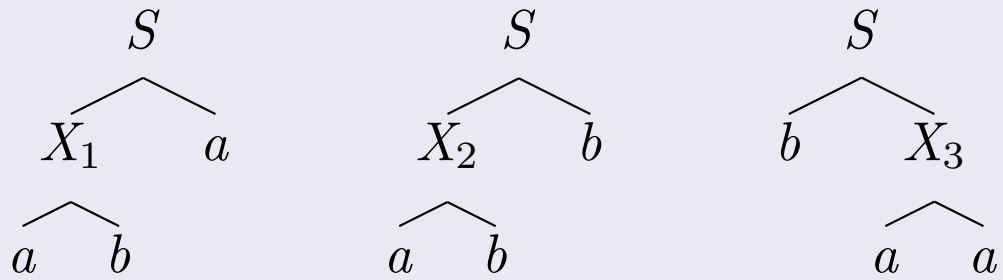
- Subclass of Deterministic CFGs.
- A substring is always a *constituent* or it is always not (it is a *distituent*).
  - For instance: $L = \{aba, abb, baa\}$



  - The constituents are $C = \{ab, aa, aba, abb, baa\}$.

UNTS Grammars: Definition

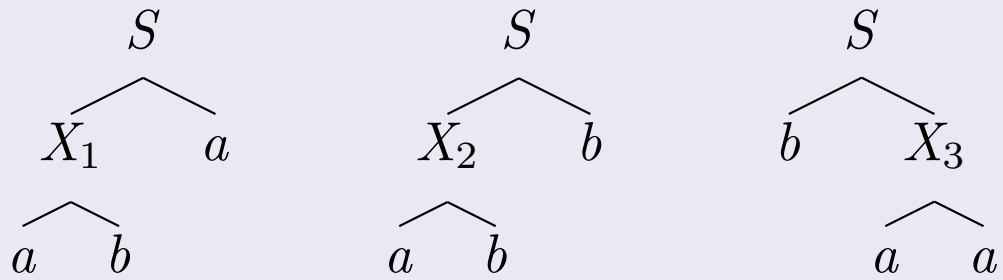
- Subclass of Deterministic CFGs.
- A substring is always a *constituent* or it is always not (it is a *distituent*).
  - For instance: $L = \{aba, abb, baa\}$



  - The constituents are $C = \{ab, aa, aba, abb, baa\}$. (this is L)

UNTS Grammars: Definition

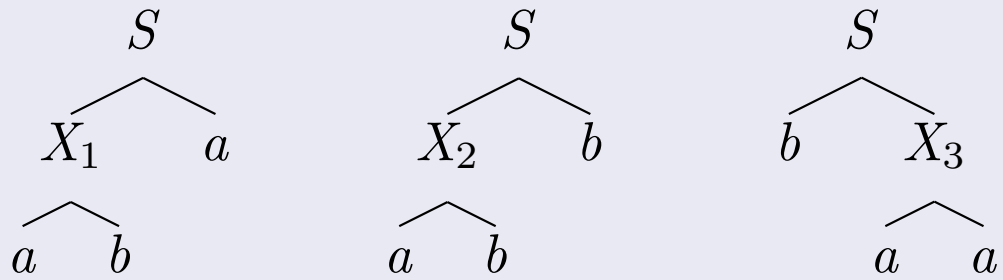
- Subclass of Deterministic CFGs.
- A substring is always a *constituent* or it is always not (it is a *distituent*).
  - For instance: $L = \{aba, abb, baa\}$



  - The constituents are $C = \{ab, aa, aba, abb, baa\}$.
  - The distituents are $D = \{ba, bb\}$. (find them!)

UNTS Grammars: Definition

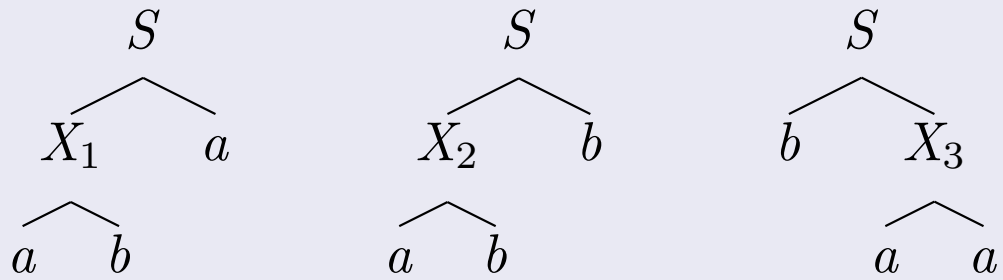
- Subclass of Deterministic CFGs.
- A substring is always a *constituent* or it is always not (it is a *distituent*).
  - For instance: $L = \{aba, abb, baa\}$



  - The constituents are $C = \{ab, aa, aba, abb, baa\}$.
  - The distituents are $D = \{ba, bb\}$.
- If two strings *overlap* and *occur overlapped* in L, both can not belong to C. We call them *incompatible*.

UNTS Grammars: Definition

- Subclass of Deterministic CFGs.
- A substring is always a *constituent* or it is always not (it is a *distituent*).
  - For instance: $L = \{aba, abb, baa\}$



  - The constituents are $C = \{ab, aa, aba, abb, baa\}$.
  - The distituents are $D = \{ba, bb\}$.
- If two strings *overlap* and *occur overlapped* in L, both can not belong to C. We call them *incompatible*.
  - For instance: ab and ba occur overlapped in aba, so they are incompatible.

UNTS Grammars: Properties

Context

- We parse a finite set of sentences (the gold treebank's).
- We ignore non-terminal labels (using unlabeled F1).

UNTS Grammars: Properties

Context

- We parse a finite set of sentences (the gold treebank's).
- We ignore non-terminal labels (using unlabeled F1).

Properties

- The possible unlabeled finite UNTS treebanks are fully characterized by its' sets of constituents C.
- The sets C are subsets of substrings of the sentences of the treebank. This is the search space.
- All the strings that belong to a set of constituents must be compatible with each other. This is, C must be *compatible*.
- We call T_C the UNTS treebank generated by a set C.

Finding the Best UNTS Grammar: Approach

- We must find a compatible set C that maximizes a score.

Finding the Best UNTS Grammar: Approach

- We must find a compatible set C that maximizes a score.
- We will use a score of the form

$$W(C) = \sum_{s \in C} w(s)$$

Finding the Best UNTS Grammar: Approach

- We must find a compatible set C that maximizes a score.
- We will use a score of the form

$$W(C) = \sum_{s \in C} w(s)$$

- $w(s)$ is the score of an individual substring s.

Finding the Best UNTS Grammar: Approach

- We must find a compatible set C that maximizes a score.
- We will use a score of the form

$$W(C) = \sum_{s \in C} w(s)$$

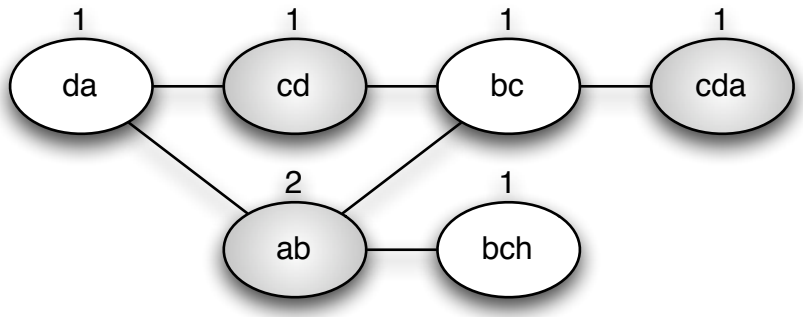
- $w(s)$ is the score of an individual substring s.
- This approach reduces the problem to the Maximum Weight Independent Set graph problem, that is known to be NP-Hard.

Finding the Best UNTS Grammar: Approach

- We must find a compatible set C that maximizes a score.
- We will use a score of the form

$$W(C) = \sum_{s \in C} w(s)$$

- $w(s)$ is the score of an individual substring s.
- This approach reduces the problem to the Maximum Weight Independent Set graph problem, that is known to be NP-Hard.



Finding the Best UNTS Grammar: The W Measure

$$W(C) = \sum_{s \in C} w(s)$$

- We must define $w(s)$.

Finding the Best UNTS Grammar: The W Measure

$$W(C) = \sum_{s \in C} w(s)$$

- We must define $w(s)$.
- $F1$ can not be written this way. We will use another measure.

Finding the Best UNTS Grammar: The W Measure

$$W(C) = \sum_{s \in C} w(s)$$

- We must define $w(s)$.
- $F1$ can not be written this way. We will use another measure.
- Let
  - $c(s)$: # of times s is a constituent in the gold treebank.
  - $d(s)$: # of times s is a distituent in the gold treebank.

Finding the Best UNTS Grammar: The W Measure

$$W(C) = \sum_{s \in C} w(s)$$

- We must define $w(s)$.
- $F1$ can not be written this way. We will use another measure.
- Let
  - $c(s)$: # of times s is a constituent in the gold treebank.
  - $d(s)$: # of times s is a distituent in the gold treebank.
- If $s \in C$ then T_C has
  - correctly tagged s $c(s)$ times (*hits*),
  - and incorrectly tagged s $d(s)$ times (*misses*).

Finding the Best UNTS Grammar: The W Measure

$$W(C) = \sum_{s \in C} w(s)$$

- We must define $w(s)$.
- $F1$ can not be written this way. We will use another measure.
- Let
  - $c(s)$: # of times s is a constituent in the gold treebank.
  - $d(s)$: # of times s is a distituent in the gold treebank.
- If $s \in C$ then T_C has
  - correctly tagged s $c(s)$ times (*hits*),
  - and incorrectly tagged s $d(s)$ times (*misses*).
- We simultaneously maximize the hits and minimize the misses.

Finding the Best UNTS Grammar: The W Measure

$$W(C) = \sum_{s \in C} c(s) - d(s)$$

- We must define $w(s)$.
- $F1$ can not be written this way. We will use another measure.
- Let
  - $c(s)$: # of times s is a constituent in the gold treebank.
  - $d(s)$: # of times s is a distituent in the gold treebank.
- If $s \in C$ then T_C has
  - correctly tagged s $c(s)$ times (*hits*),
  - and incorrectly tagged s $d(s)$ times (*misses*).
- We simultaneously maximize the hits and minimize the misses.
- So, we define the W measure with $w(s) = c(s) - d(s)$.

Finding the Best UNTS Grammar: The W Measure

$$W(C) = H(C) - M(C)$$

- We must define $w(s)$.
- $F1$ can not be written this way. We will use another measure.
- Let
  - $c(s)$: # of times s is a constituent in the gold treebank.
  - $d(s)$: # of times s is a distituent in the gold treebank.
- If $s \in C$ then T_C has
  - correctly tagged s $c(s)$ times (*hits*),
  - and incorrectly tagged s $d(s)$ times (*misses*).
- We simultaneously maximize the hits and minimize the misses.
- So, we define the W measure with $w(s) = c(s) - d(s)$.
- We also define $H(C) = \sum_{s \in C} c(s)$ and $M(C) = \sum_{s \in C} d(s)$.

Bounding the F1

Introduction

- We have a method to find the grammar with optimal W.
- We must relate W to $F1$ and find a way to bound $F1$.

Bounding the F1

Introduction

- We have a method to find the grammar with optimal W.
- We must relate W to $F1$ and find a way to bound $F1$.

Relating W and $F1$

- Write P and R in terms of H (hits) and M (misses):

$$P(C) = \frac{H(C)}{H(C)+M(C)}$$
$$R(C) = \frac{H(C)}{K}$$

- Write W in terms of precision and recall:

$$W(p, r) = (2 - \frac{1}{p})rK$$

Bounding the F1: The Bounds

A First Upper Bound

- Write $F1$ in terms of W and recall:

$$F1(r, w) = \frac{2r}{1 + 2r - \frac{w}{K}}$$

- $F1$ is monotonically increasing in r and w.
- If w is a bound of W, $F1(1.0, w)$ is a bound of $F1$.

Bounding the F1: The Bounds

A First Upper Bound

- Write $F1$ in terms of W and recall:

$$F1(r, w) = \frac{2r}{1 + 2r - \frac{w}{K}}$$

- $F1$ is monotonically increasing in r and w.
- If w is a bound of W, $F1(1.0, w)$ is a bound of $F1$.

An Improved Upper Bound (not in the paper)

- If r is a bound of R, $F1(r, w)$ is a bound of $F1$.
- With $w(s) = c(s)$ the MWIS problem optimizes recall.

Bounds for the WSJ10: Introduction

A UNTS variant

- Not any set of sentences can be generated by strict UNTS. For instance: $\{ab, bc, abc\}$ and the WSJ10 sentences.
- We use a variant of UNTS that can generate any set.
  - Whole sentences are not counted as constituents.
  - Anyway, every sentence of the corpus is accepted.
- All the bounding method applies also to the UNTS variant.

Bounds for the WSJ10: Introduction

A UNTS variant

- Not any set of sentences can be generated by strict UNTS. For instance: $\{ab, bc, abc\}$ and the WSJ10 sentences.
- We use a variant of UNTS that can generate any set.
  - Whole sentences are not counted as constituents.
  - Anyway, every sentence of the corpus is accepted.
- All the bounding method applies also to the UNTS variant.

ILP Reduction

- The MWIS problem is translated to an Integer Linear Programming problem.
- The optimals W and recall are found using ILP software (SCIP, http://scip.zib.de).

Bounds for the WSJ10: Experiments and Results

The WSJ10 Treebank

- The sentences of length ≤ 10 of the WSJ Penn Treebank.
- Used to test sate-of-the-art unsupervised parsers.
- We use as alphabet the POS tags.

Bounds for the WSJ10: Experiments and Results

The WSJ10 Treebank

- The sentences of length ≤ 10 of the WSJ Penn Treebank.
- Used to test sate-of-the-art unsupervised parsers.
- We use as alphabet the POS tags.

Experiments

- Obtain BestW using ILP.

Results

Model	UP	UR	F1
RBranch	55.1	70.0	61.7
DMV+CCM	69.3	88.0	77.6
U-DOP	70.8	88.2	78.5
UML-DOP			82.9
Incremental	75.6	76.2	75.9
BestW	91.2	62.8	**74.4**
UBoundF1	69.8	100.0	**82.2**
BestR	74.1	**69.0**	71.4
UBoundF1'	85.0	69.0	**76.1**

Bounds for the WSJ10: Experiments and Results

The WSJ10 Treebank

- The sentences of length ≤ 10 of the WSJ Penn Treebank.
- Used to test sate-of-the-art unsupervised parsers.
- We use as alphabet the POS tags.

Experiments

- Obtain BestW using ILP.
- Compute UBoundF1 in terms of BestW.

Results

Model	UP	UR	F1
RBranch	55.1	70.0	61.7
DMV+CCM	69.3	88.0	77.6
U-DOP	70.8	88.2	78.5
UML-DOP			82.9
Incremental	75.6	76.2	75.9
BestW	91.2	62.8	**74.4**
UBoundF1	69.8	100.0	**82.2**
BestR	74.1	**69.0**	71.4
UBoundF1'	85.0	69.0	**76.1**

Bounds for the WSJ10: Experiments and Results

The WSJ10 Treebank

- The sentences of length ≤ 10 of the WSJ Penn Treebank.
- Used to test sate-of-the-art unsupervised parsers.
- We use as alphabet the POS tags.

Experiments

- Obtain BestW using ILP.
- Compute UBoundF1 in terms of BestW.
- Obtain BestR using ILP.

Results

Model	UP	UR	F1
RBranch	55.1	70.0	61.7
DMV+CCM	69.3	88.0	77.6
U-DOP	70.8	88.2	78.5
UML-DOP			82.9
Incremental	75.6	76.2	75.9
BestW	91.2	62.8	**74.4**
UBoundF1	69.8	100.0	**82.2**
BestR	74.1	**69.0**	71.4
UBoundF1'	85.0	69.0	**76.1**

Bounds for the WSJ10: Experiments and Results

The WSJ10 Treebank

- The sentences of length ≤ 10 of the WSJ Penn Treebank.
- Used to test sate-of-the-art unsupervised parsers.
- We use as alphabet the POS tags.

Experiments

- Obtain BestW using ILP.
- Compute UBoundF1 in terms of BestW.
- Obtain BestR using ILP.
- Compute UBoundF1' in terms of BestW and BestR.

Results

Model	UP	UR	F1
RBranch	55.1	70.0	61.7
DMV+CCM	69.3	88.0	77.6
U-DOP	70.8	88.2	78.5
UML-DOP			82.9
Incremental	75.6	76.2	75.9
BestW	91.2	62.8	**74.4**
UBoundF1	69.8	100.0	**82.2**
BestR	74.1	**69.0**	71.4
UBoundF1'	85.0	69.0	**76.1**

Conclusions

We showed...

- a method to assess the potential of UNTS grammars over given treebanks by giving an upper bound for the F1.
- a measure W that is related to F1 and is more tractable.
- that UNTS grammars over POS tags can not parse the WSJ10 with an F1 over 76.1%.

Conclusions

We showed...

- a method to assess the potential of UNTS grammars over given treebanks by giving an upper bound for the F1.
- a measure W that is related to F1 and is more tractable.
- that UNTS grammars over POS tags can not parse the WSJ10 with an F1 over 76.1%.

But...

- the method is too computationally complex to be applied to big treebanks.
- the method applies only to UNTS and small variants.

Further Work

Already done:

- Find bounds for the UNTS-SC class (subclass of UNTS).
- Prove that W optimization in UNTS is NP-hard.

Further Work

Already done:

- Find bounds for the UNTS-SC class (subclass of UNTS).
- Prove that W optimization in UNTS is NP-hard.

To do:

- Study expresiveness of ILP to encode other classes of grammars.
- Define a logic language to express classes of grammars.
- Compute upper bounds for several known classes.

Bibliography


Rens Bod. 2006. Unsupervised parsing with U-DOP.
In *Proceedings of CoNLL-X*.


Alexander Clark. 2006. PAC-learning unambiguous NTS languages.
In *Proceedings of ICGI-2006*.


Richard M. Karp. 1972. Reducibility among combinatorial problems.
In *Complexity of Computer Computations*. Plenum Press.


Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency.
In *Proceedings of ACL 42*.


Yoav Seginer. 2007. Fast unsupervised incremental parsing.
In *Proceedings of ACL 45*.