

## Learning Computational Grammars

John Nerbonne Groningen	Anja Belz SRI Cambridge	Nicola Cancedda Xerox
Hervé Déjean Tübingen & Xerox	James Hammerton UCD Dublin	Rob Koeling SRI Cambridge
Stasinos Konstantopoulos Groningen	Miles Osborne Groningen	Franck Thollard Tübingen
	Erik Tjong Kim Sang Antwerp	

Conference on Natural Language Learning 2001  
Toulouse, 7 July 2001

## **Learning Computational Grammars (LCG)**

- Introduction, Background, People
- Scientific Motives, Goals & Objectives
- Chunking through Theory Refinement (Déjean, Tübingen)
- Memory-Based Identification of Hierarchy (Tjong Kim Sang, Antwerp)
- Overview and Comparison of Results
- Future

## Introduction

LCG is a postdoc network sponsored by the European Union's *Training and Mobility of Researchers* (TMR) program, division of Mathematical and Information Science.

Seven sites, three postdoc years each, 4/1998 - 3/2002 (some realized by graduate students).

Focus: applying machine learning techniques to natural language syntax.

TMR program emphasizes collaboration among laboratories (not only coordinators and project employees), and this has materialized

TMR program promotes collaboration with industry.

<b>Site</b>	<b>Coordinator</b>	<b>Researchers</b>
Groningen	John Nerbonne	Miles Osborne Stasinou Konstantopoulos Susanne Schoof
Tübingen	Erhard Hinrichs Dale Gerdemann	Herve Déjean Franck Thollard
Antwerp	Walter Daelemans	Erik Tjong Kim Sang
Dublin	Ronan Reilly	James Hammerton
SRI Cambridge	David Milward	Anja Belz Rob Koeling
Xerox	Christer Samuelsson Eric Gaussier	Nicola Cancedda
ISSCO, Geneva	Susan Armstrong	Adelina Hild Alexander Clark

## Background & Motivation

1983-1992 NLP

- knowledge-based processing
- careful grammar development
- deep analysis in limited domains

1992-1998

- availability of large corpora
- statistical approaches
- shallow analysis in broader domains

LCG perspective: statistical approaches apply *learning* (ML) to NLP

# ML Applied to NLP

## Scientific and Technical Motivation

- problem:
  - NLP systems need improvement
  - narrow & deep OR broad & shallow
- opportunity
  - preconditions (annotated data) available
  - little systematic work had been done

Linguistic Heritage: language acquisition as central challenge

## Goals → Objectives

General Goal: How is ML best applied in NLP?

Plan: compare several ML approaches on a single, feasible, and useful task

state of art (ca. 1997):

- tagging — assign category to word
- chunking — recognize simplest noun phrases (NP)

task: spotting and assign structure to basic phrases

- feasible — increment on “solved” problems
- useful — recognize simplest phrases
- challenging — coordination, iteration/recursion, long-distance dependency

## Task: General Text Chunking

Text chunks: non-overlapping phrases with syntactically related words

[<sub>NP</sub> He ] [<sub>VP</sub> reckons ] [<sub>NP</sub> the current account deficit ] [<sub>VP</sub> will narrow ] [<sub>P</sub> to ] [<sub>NP</sub> only £ 1.8 billion ] [<sub>P</sub> in ] [<sub>NP</sub> September ] .

eight chunks: four NP chunks, two VP chunks and two prep. chunks.

CoNLL-2000 (Lisbon) shared task  
WSJ data, definitions available at

<http://lcg-www.uia.ac.be/con112000/chunking/>

## Task: NP Chunking

NP chunking (= base NP identification)

Church 1988, Ramshaw & Marcus 1995

WSJ data, definitions available at

<http://lcg-www.uia.ac.be/~erikt/research/np-chunking.html>

## Task: General NP Bracketing

beyond chunks:

*In early trading in Hong Kong Monday , gold was quoted at \$ 366.50  
an ounce .*

NL [NP \$ 366.50 an ounce ] contains two NP's:  
[NP \$ 366.50 ] and [NP an ounce ]

NP BRACKETING: find all noun phrases in a text

Data, definitions agreed on for CoNLL-1999

See <http://lcg-www.uia.ac.be/con1199/npb/>

... and of course clause identification, CoNLL-2001!

---

## Learning NP Syntax

Site	Researchers	Learning Technique
Groningen	Miles Osborne	Minimum Description Length
Tübingen	Stasinos Konstantopoulos Herve Déjean	Inductive Logic Programming Theory Refinement (ALLiS)
Antwerp	Franck Thollard	Automaton Induction
Dublin	Erik Tjong Kim Sang	Memory-Based Learning
SRI Cambridge	James Hammerton	Neural Networks (SRN's, etc.
ISSCO, Geneva	Anja Belz	Local Structural Content
	Rob Koeling	Maximum Entropy Modeling
	Alexander Clark	Context Distribution Clusterin

## Other LCG Work

<b>Site</b>	<b>Researcher</b>	<b>Work</b>
Xerox	Nicola Cancedda	Explanation-Based Learning (Grammar Specialization)
Geneva	Adelina Hild	Corpus Analysis & Annotation
Antwerp	Erik Tjong Kim Sang	Phonotactics
Groningen	Stasinos Konstantopoulos	Phonotactics

## Open Style

three open calls for participation

- Bergen EACL 1999: NP identification
- Lisbon CoNLL 2000: general chunking
- Toulouse ACL/EACL 2001: clausing

lots of external speakers: Peter Culicover, Stephen Muggleton, Ted Briscoe, Lance Ramshaw, Marshall Maybury, Colin de la Higuera & Rens Bod.

lots of data sharing, common exploration

# CHUNKING

Hervé Déjean  
Seminar fuer Sprachwissenschaft  
Universitaet Tuebingen  
&  
XRCE - Grenoble  
*Herve.Dejean@xrce.xerox.com*

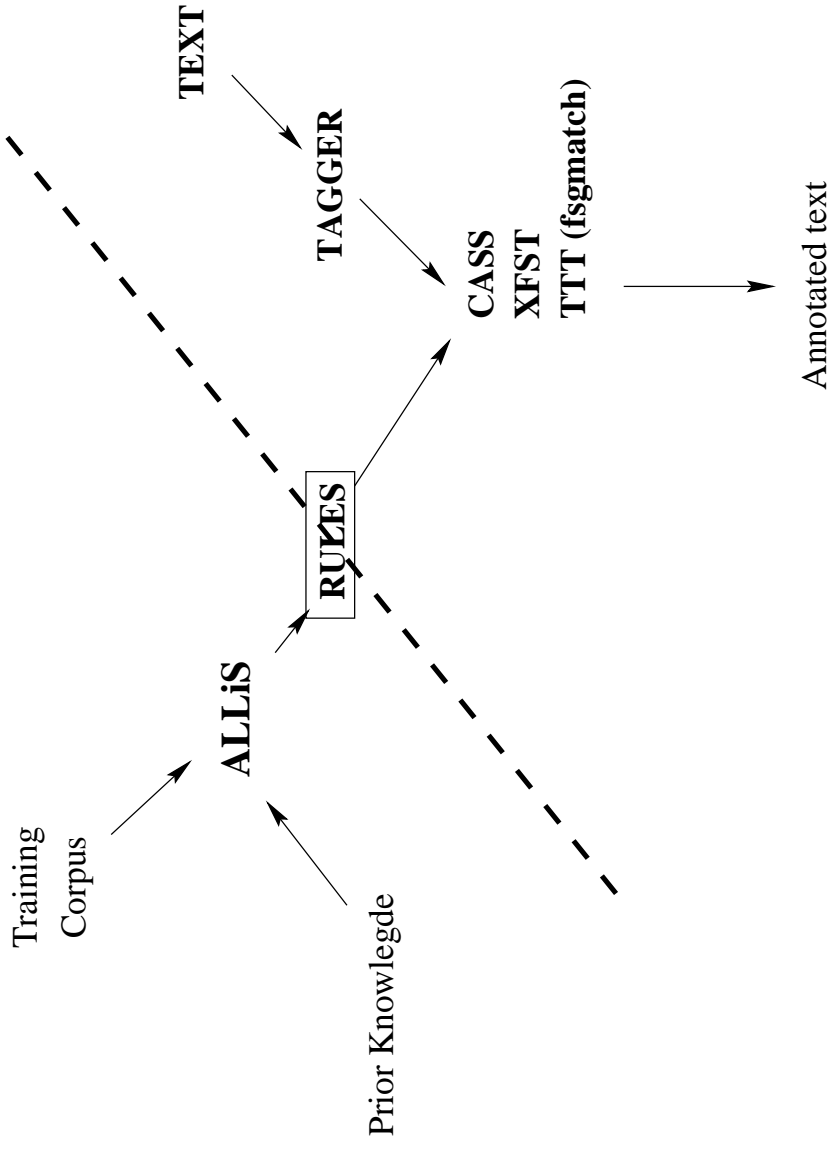
## Goal

[New/NNP York/NNP City/NNP bonds/NNS] [were/VBD sold/VBN off/IN]  
 [by/IN] [many/JJ investors/NNS] [last/JJ week/NN] ./.



New/NNP/NP	York/NNP	City/NNP	bonds/NNS
NP_B	NP_I	NP_I	NP_I
were/VBD	sold/VBN	off/IN	by/IN
VP_B	VP_I	VP_I	PP_B
many/JJ	investors/NNS	last/JJ	week/NN
NP_B	NP_I	NP_B	NP_I
			./.
			O

# ALLiS



## Motivations

- Are rule-based systems competitive wrt probabilistic approaches?
- How sophisticated the system should be?
- What kind of prior is useful?

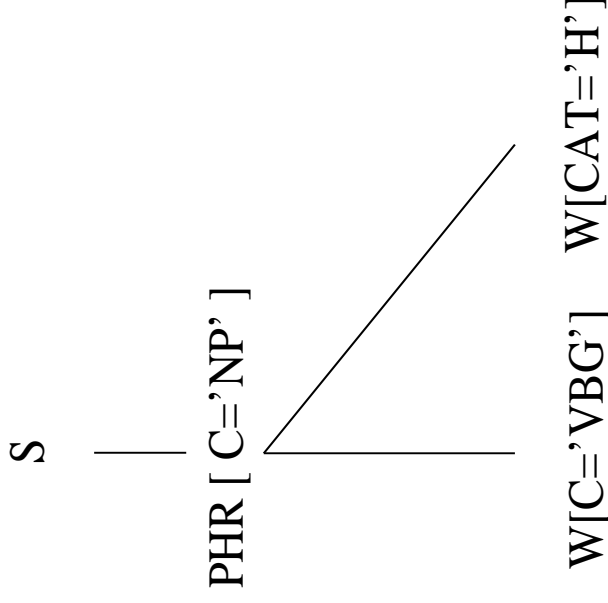
## ALLiS: the learning algorithm

1. The initial grammar
  - assign to each element its default (most frequent) category
2. The refinement
  - Find revision points
  - Create possible revisions
  - Choose best revision

Default values + general-to-specific algorithm

## General-to-specific algorithm: Adding constraints

- $W[C='VGB'] \rightarrow CAT(W)='I-NP'$



## Results (CoNLL 2000)

test data	precision	recall	$F_{\beta=1}$
Kudoh and Matsumoto	93.45	93.51	93.48
Van Halteren	93.13	93.51	93.32
Tjong Kim Sang	94.04	91.00	92.50
Zhou, Tey and Su	91.99	92.25	92.12
<b>Déjean</b>	91.87	92.31	92.09
Koeling	92.08	91.86	91.97
Osborne	91.65	92.33	91.64
Veenstra and Van den Bosch	91.05	92.03	91.54
Pla, Molina and Prieto	90.63	88.25	85.76
Johansson	86.24	88.25	87.23
Vilain and Day	88.82	82.91	85.76
baseline	72.58	82.14	77.07

## Conclusions

- Theory Refinement:
  - linguistic exception
  - noise from preceding levels
- Threshold:
  - depends of the noise level
    - \* 0.9 (tagging), 0.8 (tagging, chunking)
- Parsing (application of the rules):
  - little/no impact (minor problem in this case)
- The system is competitive
- A simple implementation of a top-down induction system is enough (large space for improvement)

---

# Identifying Hierarchical Structures

Erik F. Tjong Kim Sang  
CNTS - Language Technology Group  
University of Antwerp  
Belgium  
*erikt@uia.ua.ac.be*

## Goal

Finding noun phrases and arbitrary phrases, preferably by using the results of the base noun phrase and chunking work.

## Approach

Bottom-up phrase recognition: identify phrases at one level of the tree while using the phrases found at lower levels.

## Basis

A good method for detecting base phrases.

## Algorithm

We have used the memory-based learning algorithm IB1-IG, a nearest-neighbor classifier.

Tokens have been represented by a set of features from a window of surrounding words, part-of-speech tags and chunk tags.

All training data is stored and test data is classified by taking the class of the training data item that is closest to them in the feature space.

## Evaluation measures

Phrase detection methods will be evaluated with four rates:

1. **Precision:** percentage of phrases that were found that were correct
2. **Recall:** percentage of correct phrases that were found
3. **F:** a combination of precision and recall:  
$$F = (2 * \text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$
4. **Crossing rate:** average number of found phrases per sentence that cross correct phrases (only used for full parsing)

## Tasks

### **Task: NP Parsing**

Find arbitrary noun phrases (CoNLL-1999 shared task). Training data: sections 15-18 of the WSJ part of the Penn Treebank. Test data: WSJ section 20.

### **Task 2: Clause Identification**

Find clauses (CoNLL-2001 shared task). Training data: WSJ sections 15-18. Test data: WSJ section 20.

### **Task 3: Full Parsing**

Build complete parse trees. Training data: WSJ sections 02-21. Test data: WSJ section 23.

## NP Parsing

	Precision	Recall	F
Baseline	93.24%	67.90%	78.58
NP Parser	90.00%	78.38%	83.79
Collins Parser	89.3%	90.4%	89.8

- The baseline scores have been obtained by our best NP chunker.
- The base level NPs are detected by a combination of five MBL systems; other levels use a single MBL system.
- Each test data level was processed with the corresponding training data level only.
- State-of-the-art parsers obtain up to F=90 for this task (Collins 1999, model 2, WSJ section 23).

## Clause Identification

	Precision	Recall	F
Baseline	98.44%	31.48%	47.71
Clause Parser	76.91%	60.61%	67.79
Collins Parser	89.1%	88.3%	88.7

- The baseline scores are produced by a system which puts every sentence in a single clause.
- The clause parser estimates open and close bracket positions and ties these together with heuristic rules.
- State-of-the-art parsers obtain up to  $F=89$  for this task (Collins 1999, model 2, WSJ section 23).

## Full Parsing

	Precision	Recall	F	CB
Baseline	94.15%	33.39%	49.30	0.06
Our Parser	82.34%	78.72%	80.49	1.69
Collins Parser	89.9%	89.6%	89.7	0.87

- The baseline results were obtained by our general chunker.
- The base level phrases are detected by a combination of five MBL systems; other levels use a single MBL system.
- Each test data level was processed with the corresponding training data level only, up to the maximum level of 20.
- State-of-the-art parsers obtain up to  $F=90$  for this task (Collins 2000).

## Concluding remarks

- We have examined bottom-up chunk parsers applied to NP Parsing, Clause Identification and Full Parsing.
- The chunk parsers perform reasonably but worse than state-of-the-art statistical parsers.
- The prime problems of the parsers seems to be their greedy search strategy and their inability to use information of different parsing levels at the same time.

## System Combination

naturally, as many systems were developed, thoughts turn to combination as a means of improvement

- majority choice
- voting weighted by training performance
- “stacked classifiers” —learning applied to learners
- best-N majority choice (Tjong Kim Sang et al. 2000)

in general, combinations are improvements over best systems

## Comparison—General Chunking

	precision	recall	$F_{\beta=1}$
Memory-Based	94.04%	91.00%	92.50
Theory Refinement	91.87%	92.31%	92.09
Maximum Entropy	92.08%	91.86%	91.97
MaxEnt Tagger	91.65%	92.23%	91.94
Local Structural Context Grammar	87.97%	88.17%	88.07
Automaton Induction	84.92%	86.75%	85.82
combination (majority)	93.68%	92.98%	93.33
best	93.45%	93.51%	93.48
baseline	72.58%	82.14%	77.07

no lexical information in LSCG, Automaton Induction  
 baseline — most frequent chunk tag  
 best — support vector machines (external participant)

## NP Chunking

	precision	recall	$F_{\beta=1}$
Memory-Based	93.63%	92.88%	93.25
Maximum Entropy	93.20%	93.00%	93.10
Theory Refinement	92.49%	92.69%	92.59
IGTree (Memory-Based)	92.28%	91.65%	91.96
C5.0 (Decision Tree)	89.59%	90.66%	90.12
Self-Organizing Neural Nets	89.29%	89.73%	89.51
combination (best-3 majority)	93.78%	93.52%	93.65
best	94.18%	93.55%	93.86
baseline	78.20%	81.87%	79.99

no lexical information in C5.0, Self-Organizing Nets

---

## NP Bracketing

	precision	recall	$F_{\beta=1}$
Memory-Based	90.00%	78.38%	83.79
Local Structural Context Grammar	80.04%	80.25%	80.15
Minimum Description Length	53.2%	68.7%	59.9
best	91.28%	76.06%	82.98
baseline	77.57%	59.85%	67.56

---

## Technical Conclusions

- surprising convergence in accuracy levels
- memory-based learning consistently strong
- still difficult to capitalize on extensive linguistic bias
- learning speed/capacity seems the bottleneck, e.g., for ILP, NN's
- system combination consistently useful
- incremental steps toward full parsing arrive quickly at points where full parsing is better

## Organization and Collaboration

### “Shared-Task” Paradigm

- several groups work on same task
- share data, evaluation standards
  - competitive element (within limits)
- regular comparison, exchange of ideas
- examples in CL, but also
  - Critical Assessment of Structure Prediction (CASP)  
(structural models for amino acid sequences)

## Advantages and Disadvantages

### Advantages

- each participant “solves” problem
  - rather than contributing single module
  - raising general technical level
- comparison, understanding of alternatives
- competition → rapid dissemination among participants

### Disadvantage

- competition leads to focus on numbers ( $F$ -scores)
  - built-in in order to get the most out of each method

## Scientific Highlights

3rd, 4th, 5th CoNLL's

several (near)-best results

- three of world's four best results in NP chunking (as of 12/2000)
- four of seven best results in general chunking
- NP identification / full-parse selection

several co-publications

## Future

- further comparison
- error analysis – which techniques work where?  
—e.g., 30% of NP chunking errors were annotation errors
- CFP *Journal of Machine Learning Research*
- unsupervised techniques