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.
<table>
<thead>
<tr>
<th>Site</th>
<th>Coordinator</th>
<th>Researchers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groningen</td>
<td>John Nerbonne</td>
<td>Miles Osborne</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stasinos Konstantopoulos</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Susanne Schoof</td>
</tr>
<tr>
<td>Tübingen</td>
<td>Erhard Hinrichs</td>
<td>Hérve Déjean</td>
</tr>
<tr>
<td></td>
<td>Dale Gerdemann</td>
<td>Franck Thollard</td>
</tr>
<tr>
<td>Antwerp</td>
<td>Walter Daelemans</td>
<td>Erik Tjong Kim Sang</td>
</tr>
<tr>
<td>Dublin</td>
<td>Ronan Reilly</td>
<td>James Hammerton</td>
</tr>
<tr>
<td>SRI Cambridge</td>
<td>David Milward</td>
<td>Anja Belz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rob Koeling</td>
</tr>
<tr>
<td>Xerox</td>
<td>Christer Samuelsson</td>
<td>Nicola Cancendedda</td>
</tr>
<tr>
<td></td>
<td>Eric Gaussier</td>
<td></td>
</tr>
<tr>
<td>ISSCO, Geneva</td>
<td>Susan Armstrong</td>
<td>Adelina Hild</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Alexander Clark</td>
</tr>
</tbody>
</table>
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 \rightarrow 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

\[ [\text{NP He} \quad [\text{VP reckons} \quad [\text{NP the current account deficit} \quad [\text{VP will narrow} \quad [\text{P to} \quad [\text{NP only £ 1.8 billion} \quad [\text{P in} \quad [\text{NP September} \quad ] \quad ] \quad ] \quad ] \quad ] \quad ] \quad ] \quad . \]

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

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

Task: NP Chunking

NP chunking (= base NP identification)
Church 1988, Ramshaw & Marcus 1995
WSJ data, definitions available at

Task: General NP Bracketing

beyond chunks:

_In early trading in Hong Kong Monday, gold was quoted at $366.50 an ounce._

NL \([\text{NP } \$ 366.50 \text{ an ounce}]\) contains two NP’s: 
\([\text{NP } \$ 366.50]\) and \([\text{NP \ an ounce}]\)

NP Bracketing: find all noun phrases in a text

Data, definitions agreed on for CoNLL-1999

... and of course clause identification, CoNLL-2001!
# Learning NP Syntax

<table>
<thead>
<tr>
<th>Site</th>
<th>Researchers</th>
<th>Learning Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groningen</td>
<td>Miles Osborne, Stasinos Konstantopoulos</td>
<td>Minimum Description Length</td>
</tr>
<tr>
<td>Tübingen</td>
<td>Hérve Déjean, Franck Thollard</td>
<td>Inductive Logic Programming</td>
</tr>
<tr>
<td>Antwerp</td>
<td>Erik Tjong Kim Sang</td>
<td>Theory Refinement (ALLiS)</td>
</tr>
<tr>
<td>Dublin</td>
<td>James Hammerton</td>
<td>Automaton Induction</td>
</tr>
<tr>
<td>SRI Cambridge</td>
<td>Anja Belz, Rob Koeling</td>
<td>Memory-Based Learning</td>
</tr>
<tr>
<td>ISSCO, Geneva</td>
<td>Alexander Clark</td>
<td>Neural Networks (SRN’s, etc.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local Structural Content</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum Entropy Modeling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Context Distribution Clustering</td>
</tr>
</tbody>
</table>
## Other LCG Work

<table>
<thead>
<tr>
<th>Site</th>
<th>Researcher</th>
<th>Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xerox</td>
<td>Nicola Cancedda</td>
<td>Explanation-Based Learning (Grammar Specialization)</td>
</tr>
<tr>
<td>Geneva</td>
<td>Adelina Hild</td>
<td>Corpus Analysis &amp; Annotation</td>
</tr>
<tr>
<td>Antwerp</td>
<td>Erik Tjong Kim Sang</td>
<td>Phonotactics</td>
</tr>
<tr>
<td>Groningen</td>
<td>Stasinos Konstantopoulos</td>
<td>Phonotactics</td>
</tr>
</tbody>
</table>
Open Style

three open calls for participation

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


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] ./.  

\[
\begin{array}{cccc}
\text{New/NNP/NP} & \text{York/NNP} & \text{City/NNP} & \text{bonds/NNS} \\
\text{NP\_B} & \text{NP\_I} & \text{NP\_I} & \text{NP\_I} \\
\text{were/VBD} & \text{sold/VBN} & \text{off/IN} & \text{by/IN} \\
\text{VP\_B} & \text{VP\_I} & \text{VP\_I} & \text{PP\_B} \\
\text{many/JJ} & \text{investors/NNS} & \text{last/JJ} & \text{week/NN} \\
\text{NP\_B} & \text{NP\_I} & \text{NP\_B} & \text{NP\_I} \\
\end{array}
\]

./.
ALLiS

Training Corpus

Prior Knowledge

RULES

CASS
XFST
TTT (fsgmatch)

Annotated text

TEXT

TAGGER
Motivations

- Are rule-based systems competitive \textit{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

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

```
S
  |  
  PHR [ C='NP' ]
   |  
  W[C='VBG']  W[CAT='H']
```
# Results (CoNLL 2000)

<table>
<thead>
<tr>
<th>test data</th>
<th>precision</th>
<th>recall</th>
<th>$F_{\beta=1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kudoh and Matsumoto</td>
<td>93.45</td>
<td>93.51</td>
<td>93.48</td>
</tr>
<tr>
<td>Van Halteren</td>
<td>93.13</td>
<td>93.51</td>
<td>93.32</td>
</tr>
<tr>
<td>Tjong Kim Sang</td>
<td>94.04</td>
<td>91.00</td>
<td>92.50</td>
</tr>
<tr>
<td>Zhou, Tey and Su</td>
<td>91.99</td>
<td>92.25</td>
<td>92.12</td>
</tr>
<tr>
<td>Déjean</td>
<td>91.87</td>
<td>92.31</td>
<td>92.09</td>
</tr>
<tr>
<td>Koeling</td>
<td>92.08</td>
<td>91.86</td>
<td>91.97</td>
</tr>
<tr>
<td>Osborne</td>
<td>91.65</td>
<td>92.33</td>
<td>91.64</td>
</tr>
<tr>
<td>Veenstra and Van den Bosch</td>
<td>91.05</td>
<td>92.03</td>
<td>91.54</td>
</tr>
<tr>
<td>Pla, Molina and Prieto</td>
<td>90.63</td>
<td>88.25</td>
<td>85.76</td>
</tr>
<tr>
<td>Johansson</td>
<td>86.24</td>
<td>88.25</td>
<td>87.23</td>
</tr>
<tr>
<td>Vilain and Day</td>
<td>88.82</td>
<td>82.91</td>
<td>85.76</td>
</tr>
<tr>
<td>baseline</td>
<td>72.58</td>
<td>82.14</td>
<td>77.07</td>
</tr>
</tbody>
</table>
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@via.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:
   \[ F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]

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


Task 3: Full Parsing

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

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>93.24%</td>
<td>67.90%</td>
<td>78.58</td>
</tr>
<tr>
<td>NP Parser</td>
<td>90.00%</td>
<td>78.38%</td>
<td>83.79</td>
</tr>
<tr>
<td>Collins Parser</td>
<td>89.3%</td>
<td>90.4%</td>
<td>89.8</td>
</tr>
</tbody>
</table>

- 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).
Claude Identification

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>98.44%</td>
<td>31.48%</td>
<td>47.71</td>
</tr>
<tr>
<td>Clause Parser</td>
<td>76.91%</td>
<td>60.61%</td>
<td>67.79</td>
</tr>
<tr>
<td>Collins Parser</td>
<td>89.1%</td>
<td>88.3%</td>
<td>88.7</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F</th>
<th>CB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>94.15%</td>
<td>33.39%</td>
<td>49.30</td>
<td>0.06</td>
</tr>
<tr>
<td>Our Parser</td>
<td>82.34%</td>
<td>78.72%</td>
<td>80.49</td>
<td>1.69</td>
</tr>
<tr>
<td>Collins Parser</td>
<td>89.9%</td>
<td>89.6%</td>
<td>89.7</td>
<td>0.87</td>
</tr>
</tbody>
</table>

- 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

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

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_{\beta=1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory-Based</td>
<td>93.63%</td>
<td>92.88%</td>
<td>93.25</td>
</tr>
<tr>
<td>Maximum Entropy</td>
<td>93.20%</td>
<td>93.00%</td>
<td>93.10</td>
</tr>
<tr>
<td>Theory Refinement</td>
<td>92.49%</td>
<td>92.69%</td>
<td>92.59</td>
</tr>
<tr>
<td>IGTee (Memory-Based)</td>
<td>92.28%</td>
<td>91.65%</td>
<td>91.96</td>
</tr>
<tr>
<td>C5.0 (Decision Tree)</td>
<td>89.59%</td>
<td>90.66%</td>
<td>90.12</td>
</tr>
<tr>
<td>Self-Organizing Neural Nets</td>
<td>89.29%</td>
<td>89.73%</td>
<td>89.51</td>
</tr>
<tr>
<td>combination (best-3 majority)</td>
<td>93.78%</td>
<td>93.52%</td>
<td>93.65</td>
</tr>
<tr>
<td>best</td>
<td>94.18%</td>
<td>93.55%</td>
<td>93.86</td>
</tr>
<tr>
<td>baseline</td>
<td>78.20%</td>
<td>81.87%</td>
<td>79.99</td>
</tr>
</tbody>
</table>

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

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