# Syntactic Formalisms for Parsing Natural Languages

Aleš Horák, Miloš Jakubíček, Vojtěch Kovář
(based on slides by Juyeon Kang)

ia161@nlp.fi.muni.cz

Autumn 2013

## Study materials

Course materials and homeworks are available on the following web site

`https://is.muni.cz/course/fi/autumn2011/IA161`

# Outline

- Introduction to Statistical parsing methods
- Statistical Parsers
  - RASP system
  - Stanford parser
  - Collins parser
  - Charniak parser
  - Berkeley parser

# 1. Introduction to statistical parsing

- The main theoretical approaches behind modern statistical parsers
- Over the last 12 years statistical parsing has succeeded significantly!
- NLP researchers have produced a range of statistical parsers

$\rightarrow$ wide-coverage and robust parsing accuracy

- They continues to improve the parsers year on year.

# Application domains of statistical parsing

- Question answering systems of high precision
- Named entity extraction
- Syntactically based sentence compressions
- Extraction of people’s opinion about products
- Improved interaction in computer ganes
- Helping linguists find data

# NLP parsing problem and solution

- The structure of language is ambiguous!

→ local and global ambiguities

- **Classical parsing problem**

→ simple 10 grammar rules can generate 592 parsers

→ real size wide-coverage grammar generates millions of parses

# NLP parsing problem and solution

## NLP parsing solution

We need mechanisms that allow us to find the most likely parses

$\rightarrow$ statistical parsing lets us work with very loose grammars that admit millions of parses for sentences but to still quickly find the best parses

# Improved methodology for robust parsing

## The annotated data: Penn Treebank (early 90’s)

- Building a treebank seems a lot slower and less useful than building a grammar
- But it has many helpful things
  - Reusability of the labor
  - Broad coverage
  - Frequencies and distributional information
  - A way to evaluate systems

# Characterization of Statistical parsing

- What the grammar which determines the set of legal syntactic structures for a sentence? How is that grammar obtained?
- What is the algorithm for determining the set of legal parses for a sentence?
- What is the model for determining the probability of different parses for a sentence?
- What is the algorithm, given the model and a set of possible parses which finds the best parse?

## Characterization of Statistical parsing

$$T_{\text{best}} = \text{arg max } Score(T, S)$$

Two components:

- The **model**: a function Score which assigns scores (probabilities) to tree and sentence pairs
- The **parser**: the algorithm which implements the search for $T_{\text{best}}$

## Characterization of Statistical parsing

Statistical parsing seen as more of a **pattern recognition**/**Machine Learning** problem plus search

The grammar is only implicitly defined by the training data and the method used by the parser for generating hypotheses

# Statistical parsing models

Probabilistic approach would suggest the following for the Score function

$$Score(T, S) = P(T|S)$$

Lots of research on different probability models for Penn Treebank trees

- Generative models, log-linear (maximum entropy) models, ...

## 2. Statistical parsers

- Many kinds of parsers based on the statistical methods:probability, machine learning
- Different objectives: research, commercial, pedagogical
  - RASP, Stanford parser, Berkeley parser,

# RASP system

**Robust Accurate Statistical Parsing (2nd release):**
[Briscoe&Carroll, 2002; Briscoe *et al.* 2006]

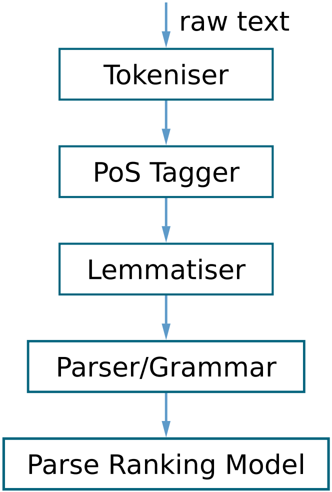
- system for syntactic annotation of free text
- Semantically-motivated output representation
- Enhanced grammar and part-of-speech tagger lexicon
- Flexible and semi-supervised training method for structural parse ranking model

Useful links to RASP

`http://ilexir.co.uk/applications/rasp/download/`

`http://www.informatics.susx.ac.uk/research/groups/nlp/rasp/`

# Components of system




- Input:

unannotated text or transcribed (and punctuated) speech

- $1^{st}$ step:

sentence boundary detection and tokenisation modules

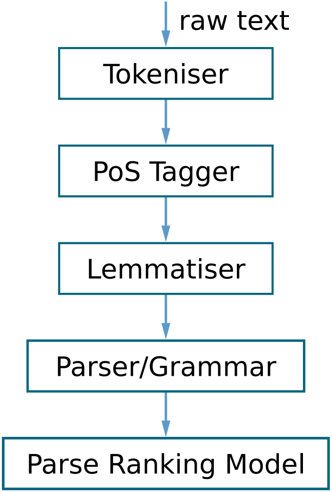
- $2^{nd}$ step:

Tokenized text is tagged with one of 150 POS and punctuation labels (derived from the CLAWS tagset)

$\rightarrow$ first-order ('bigram') HMM tagger

$\rightarrow$ trained on the manually corrected tagged version of the Susanne, LOB and BNC corpora

# Components of system




- 3rd step:

Morphological analyzer

- 4th step:

Manually developed wide-coverage tag sequence grammar in the parser

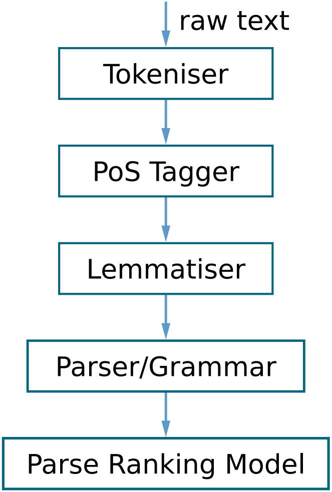
$\rightarrow$ 689 unification based phrase structure rules

$\rightarrow$ preterminals to this grammar are the POS and punctuation tags

$\rightarrow$ terminals are featural description of the preterminals

$\rightarrow$ non-terminals project information up the tree using an X-bar scheme with 41 attributes with a maximum of 33 atomic values

# Components of system




- $5^{th}$ step:

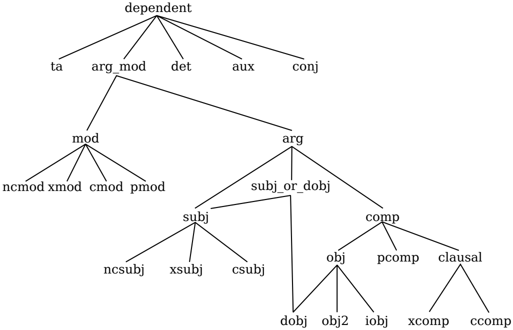
Generalized LR Parser

$\rightarrow$ a non-deterministic LALR table is constructed automatically from CF 'backbone' compiled from the featurebased grammar

$\rightarrow$ the parser builds a packed parse forest using this table to guide the actions it performs

$\rightarrow$ the n-best parses can be efficiently extracted by unpacking sub-analyses, following pointers to contained subanalyses and choosing alternatives in order of probabilistic ranking

# Components of system




- Output:

set of named grammatical relations (GRs)

→ resulting set of ranked parses can be displayed or passed on for further processing

→ transformation of derivation trees into a set of named GRs

→ GR scheme captures those aspects of predicate-argument structure

# Evaluation

- The system has been evaluated using the re-annotation of the PARC dependency bank (DepBank, King *et al.*, 2003)
- It consists of 560 sentences chosen randomly from section 23 of the WSJ with grammatical relations compatible with RASP system.
- Form of relations

(relation subtype head dependent initial)

Type of relationship between the head and the dependent

Encoding additional specifications of the relation type for some relations and the initial or underlying logical relation of the grammatical subject in constructions such as passive

# Evaluation

| Relation | Precision | Recall | $F_1$ | std GRs |
|---|---|---|---|---|
| dependent | 79.76 | 77.49 | 78.61 | 10696 |
| aux | 93.33 | 91.00 | 92.15 | 400 |
| conj | 72.39 | 72.27 | 72.33 | 595 |
| ta | 42.61 | 51.37 | 46.58 | 292 |
| det | 87.73 | 90.48 | 89.09 | 1114 |
| arg_mod | 79.18 | 75.47 | 77.28 | 8295 |
| mod | 74.43 | 67.78 | 70.95 | 3908 |
| ncmod | 75.72 | 69.94 | 72.72 | 3550 |
| xmod | 53.21 | 46.63 | 49.70 | 178 |
| cmod | 45.95 | 30.36 | 36.56 | 168 |
| pmod | 30.77 | 33.33 | 32.00 | 12 |
| arg | 77.42 | 76.45 | 76.94 | 4387 |
| subj_or_dobj | 82.36 | 74.51 | 78.24 | 3127 |
| subj | 78.55 | 66.91 | 72.27 | 1363 |
| ncsubj | 79.16 | 67.06 | 72.61 | 1354 |
| xsubj | 33.33 | 28.57 | 30.77 | 7 |
| csubj | 12.50 | 50.00 | 20.00 | 2 |
| comp | 75.89 | 79.53 | 77.67 | 3024 |
| obj | 79.49 | 79.42 | 79.46 | 2328 |
| dobj | 83.63 | 79.08 | 81.29 | 1764 |
| obj2 | 23.08 | 30.00 | 26.09 | 20 |
| iobj | 70.77 | 76.10 | 73.34 | 544 |
| clausal | 60.98 | 74.40 | 67.02 | 672 |
| xcomp | 76.88 | 77.69 | 77.28 | 381 |
| ccomp | 46.44 | 69.42 | 55.55 | 291 |
| pcomp | 72.73 | 66.67 | 69.57 | 26 |
| | | | | |
| macroaverage | 62.12 | 63.77 | 62.94 | |
| microaverage | 77.66 | 74.98 | 76.29 | |

Parsing accuracy on DepBank [Briscoe *et al.*, 2006]

- Micro-averaged precision, recall and $F_1$ score are calculated from the counts for all relations in the hierarchy
- Macro-averaged scores are the mean of the individual scores for each relation
- Micro-averaged $F_1$ score of 76.3% across all relations

# Stanford parser

**Java implementation of probabilistic natural language parsers (version 1.6.9)**

: [Klein and Manning, 2003]

- Parsing system for English and has been used in Chinese, German, Arabic, Italian, Bulgarian, Portuguese
- Implementation, both highly optimized PCFG and lexicalized dependency parser, and lexicalized PCFG parser
- Useful links

`http://nlp.stanford.edu/software/lex-parser.shtml`
`http://nlp.stanford.edu:8080/parser/`

# Stanford parser

- Input

various form of plain text

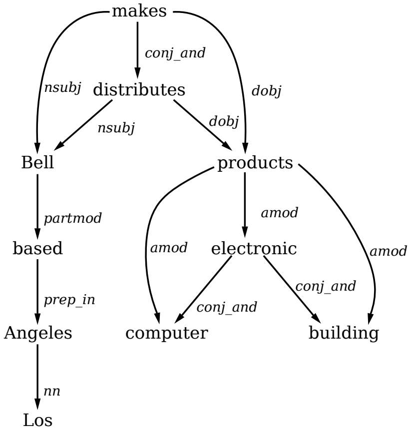
- Output

Various analysis formats

→ Stanford Dependencies (SD): typed dependencies as GRs

→ phrase structure trees

→ POS tagged text




**Graphical representation of the SD for the sentence**

"Bell, based in Los Angeles, makes and distributes electronic, computer and building products."

# Standford typed dependencies [De Marmette and Manning, 2008]

- provide a simple description of the grammatical relationships in a sentence
- represents all sentence relationships uniformly as typed dependency relations
- quite accessible to non-linguists thinking about tasks involving information extraction from text and is quite effective in relation extraction applications.

# Standford typed dependencies [De Marnette and Manning, 2008]

For an example sentence:

*Bell, based in Los Angeles, makes and distributes electronic, computer and building products.*

Stanford Dependencies (SD) representation is:

nsubj(makes-8, Bell-1)
nsubj(distributes-10, Bell-1)
partmod(Bell-1, based-3)
nn(Angeles-6, Los-5)
prep_in(based-3, Angeles-6)
root(ROOT-0, makes-8)

conj_and(makes-8, distributes-10)
amod(products-16, electronic-11)
conj_and(electronic-11, computer-13)
amod(products-16, computer-13)
conj_and(electronic-11, building-15)
amod(products-16, building-15)
dobj(makes-8, products-16)
dobj(distributes-10, products-16)

# Output

A lineup of masseurs was waiting to take the media in hand.

## POS tagged text

Parsing [sent. 4 len. 13]: [A, lineup, of, masseurs, was, waiting, to, take, the, media, in, hand, .]

## CFPSG representation

```
(ROOT
  (S
    (NP
      (NP (DT A) (NN lineup))
      (PP (IN of)
        (NP (NNS masseurs))))
    (VP (VBD was)
      (VP (VBG waiting)
        (S
          (VP (TO to)
            (VP (VB take)
              (NP (DT the) (NNS media))
              (PP (IN in)
                (NP (NN hand))))))))
    (. .)))
```

## Typed dependencies representation

```
det(lineup2, A1)
nsubj(waiting6, lineup2)
xsubj(take8, lineup2)
prep_of(lineup2, masseurs4)
aux(waiting6, was5)
root(ROOT0, waiting6)
aux(take8, to7)
xcomp(waiting6, take8)
det(media10, the9)
dobj(take8, media10)
prep_in(take8, hand12)
```

# Berkeley parser

**Learning PCFGs, statistical parser (release 1.1, version 09.2009)**

: [Petrov *et al.*, 2006; Petrov and Klein, 2007]

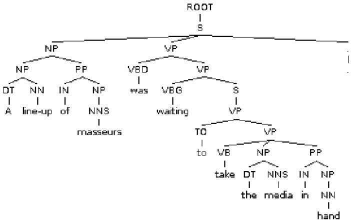
- Parsing system for English and has been used in Chinese, German, Arabic, Bulgarian, Portuguese, French
- Implementation of unlexicalized PCFG parser
- Useful links

http://nlp.cs.berkeley.edu/
http://tomato.banatao.berkeley.edu:
8080/parser/parser.html
http://code.google.com/p/berkeleyparser/

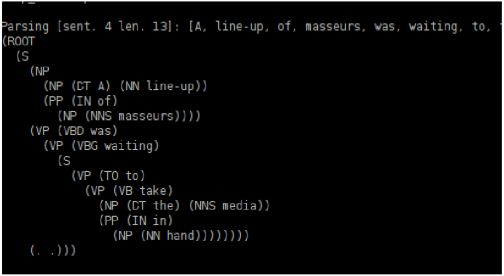
# Comparison of parsing an example sentence

A lineup of masseurs was waiting to take the media in hand.

```
(ROOT
  (S
    (NP
      (NP (DT A) (NN line-up))
      (PP (IN of)
        (NP (NNS masseurs))))
    (VP (VBD was)
      (VP (VBG waiting)
        (S
          (VP (TO to)
            (VP (VB take)
              (NP (DT the) (NNS media))
              (PP (IN in)
                (NP (NN hand))))))))
    (. .)))
```




**Berkeley parser**



```
Parsing [sent. 4 len. 13]: [A, line-up, of, masseurs, was, waiting, to,
(ROOT
  (S
    (NP
      (NP (DT A) (NN line-up))
      (PP (IN of)
        (NP (NNS masseurs))))
    (VP (VBD was)
      (VP (VBG waiting)
        (S
          (VP (TO to)
            (VP (VB take)
              (NP (DT the) (NNS media))
              (PP (IN in)
                (NP (NN hand))))))))
    (. .)))
```

**Stanford parser**

# charniak parser

**Probabilistic LFG F-Structure Parsing**
: [Charniak, 2000; Bikel, 2002]

- Parsing system for English
- PCFG based wide coverage LFG parser
- Useful links

http://nclt.computing.dcu.ie/demos.html
http://lfg-demo.computing.dcu.ie/lfgparser.html

# Collins parser

**Head-Driven Statistical Models for natural language parsing (Release 1.0, version 12.2002)**
: [Collins, 1999]

- Parsing system for English
- Useful links

`http://www.cs.columbia.edu/~mcollins/code.html`

# Bikel's parser

**Multilingual statistical parsing engine (release 1.0, version 06.2008)**
: [Charniak, 2000; Bikel, 2002]

- Parsing system for English, Chinese, Arabic, Korean

http://www.cis.upenn.edu/~dbikel/#stat-parser
http://www.cis.upenn.edu/~dbikel/software.html

# Comparing parser speed on section 23 of WSJ Penn Treebank

| Parser | Time (min.) |
|---|---|
| Collins | 45 |
| Charniak | 28 |
| Sagae | 11 |
| CCG | 1.9 |