Dependency parsing
& Dependency parsers

Lecture 11

qkang@fi.muni.cz

Syntactic formalisms for natural language parsing

IA161, FI MU autumn 2011

Study materials

- Course materials and homeworks are available on the following web site:

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

- Refer to *Dependency Parsing*, Synthesis: Lectures on Human Language Technologies, S. kübler, R. McDonald and J. Nivre, 2009

Outline

- Introduction to Dependency parsing methods
- Dependency Parsers

1. Introduction to Dependency parsing

- **Motivation**

  a. dependency-based syntactic representation seem to be useful in many applications of language technology: machine translation, information extraction

  → transparent encoding of predicate-argument structure

  b. dependency grammar is better suited than phrase structure grammar for language with free or flexible word order

  → analysis of diverse languages within a common framework

- **Motivation (Cont.)**

c. leading to the development of accurate syntactic parsers for a number of languages

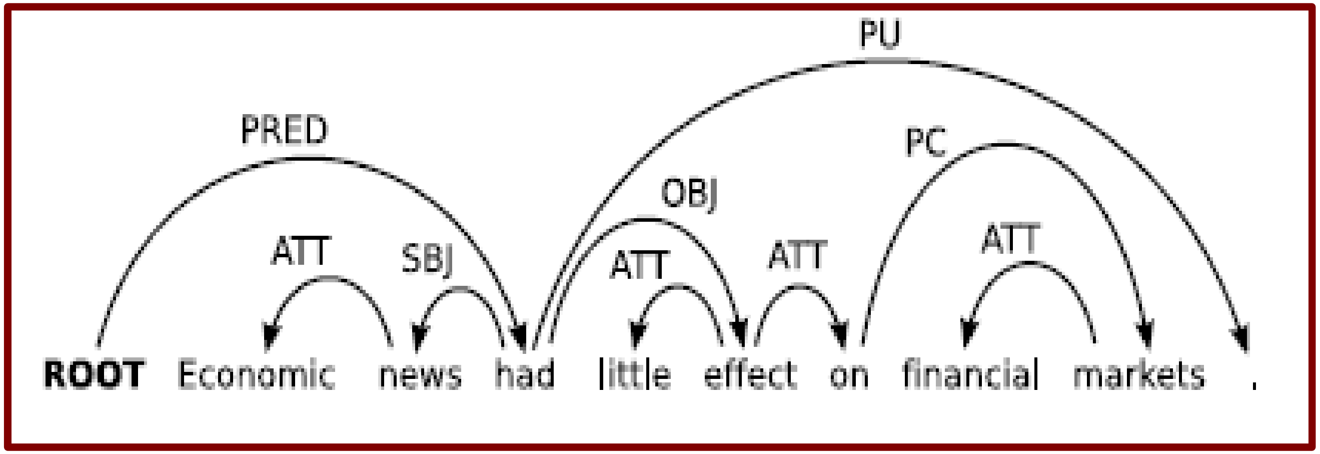
→ combination with machine learning from syntactically annotated corpora (e.g. treebank)

- **Dependency parsing**

“Task of automatically analyzing the dependency structure of a given input sentence”

- **Dependency parser**

“Task of producing a labeled dependency structure of the kind depicted in the follow figure, where the words of the sentence are connected by typed dependency relations”



Definitions of dependency graphs and dependency parsing

- **Dependency graphs**: syntactic structures over sentences

Def. 1.: A *sentence* is a sequence of tokens denoted by

$$S = w_0 w_1 \ldots w_n$$

Def. 2.: Let $R = \{r_1, \ldots, r_m\}$ be a finite set of *possible dependency relation types* that can hold between any two words in a sentence. A relation type $r \in R$ is additionally called an *arc label*.

Definitions of dependency graphs and dependency parsing (Cont.)

- **Dependency graphs**: syntactic structures over sentences

Def. 3.: A dependency graph $G=(V, A)$ is a labeled directed graph, consists of nodes, V, and arcs, A, such that for sentence $S= w_0 w_1 \ldots w_n$ and label set R the following holds:

1. $V \subseteq \{w_0 w_1 \ldots w_n\}$

2. $A \subseteq V \times R \times V$

3. if $(w_{i,} r, w_j) \in A$ then $(w_{i,} r', w_j) \notin A$ for all $r' \neq r$

- **Approach to dependency parsing**

a. data-driven

it makes essential use of machine learning from linguistic data in order to parse new sentences

b. grammar-based

it relies on a formal grammar, defining a formal language, so that it makes sense to ask whether a given input is in the language defined by the grammar or not.

→ Data-driven have attracted the most attention in recent years.

- **Data-driven approach**

according to the *type of parsing model* adopted,

the *algorithms used to learn the model from data*

the *algorithms used to parse new sentences with the model*

a. transition-based

start by defining a transition system, or state machine, for mapping a sentence to its dependency graph.

b. graph-based

start by defining a space of candidate dependency graphs for a sentence

- **Data-driven approach (Cont.)**

a. transition-based

learning problem: induce a model for predicting the next state transition, given the transition history

parsing problem: construct the optimal transition sequence for the input sentence,, given induced model

b. graph-based

learning problem: induce a model for assigning scores to the candidate dependency graphs for a sentence

parsing problem: find the highest-scoring dependency graph for the input sentence, given induced model

Transition-based Parsing

- Transition system consists of a set C of parser configurations and of a set D of transitions between configurations.

- **Main idea**: a sequence of valid transitions, starting in the *initial configuration* for a given sentence and ending in one of several *terminal configurations*, defines a valid dependency tree for the input sentence.

$$D_{1,m} = d_1(c_1),........, d_m(c_m)$$

- **Definition**

  Score of $D_{1,m}$ factors by configuration-transition pairs (c_i, d_i):

  $$s(D_{1,m}) = \Sigma_{i=1}^{m} s(c_i, d_i)$$

- **Learning**

  Scoring function $s(c_i, d_i)$ for $d_i(c_i) \in D_{1,m}$

- **Inference**

  Search for highest scoring sequence $D^*_{1,m}$ given $s(c_i, d_i)$

Transition-based Parsing (Cont.)

Inference for transition-based parsing

- **Common inference strategies:**
  - Deterministic [Yamada and Matsumoto 2003, Nivre et al. 2004]
  - Beam search [Johansson and Nugues 2006, Titov and Henderson 2007]
  - Complexity given by upper bound on transition sequence length
- **Transition system**
  - Projective O(n) [Yamada and Matsumoto 2003, Nivre 2003]
  - Limited non-projective O(n) [Attardi 2006, Nivre 2007]
  - Unrestricted non-projective O(n2) [Nivre 2008, Nivre 2009]

Transition-based Parsing (Cont.)

Learning for transition-based parsing

- **Typical scoring function:**
  - $s(c_i, d_i) = \mathbf{w} \bullet \mathbf{f}(c_i, d_i)$ where $\mathbf{f}(c_i, d_i)$ is a feature vector over configuration c_i and transition d_i and **w** is a weight vector [w_i =weight of feature f_i (c_i, d_i)]
- **Transition system**
  - Projective O(n) [Yamada and Matsumoto 2003, Nivre 2003]
  - Limited non-projective O(n) [Attardi 2006, Nivre 2007]
  - Unrestricted non-projective O(n2) [Nivre 2008, Nivre 2009]
- **Problem**
  - Learning is local but features are based on the global history

Graph-based Parsing

- For a input sentence S we define a graph $G_s=(V_s, A_s)$ where

  $V_s = \{w_0, w_1, ..., w_n\}$ and

  $A_s = \{(w_i, w_j, l) \mid w_i, w_j \in V \text{ and } l \in L\}$
- Score of a dependency tree T factors by subgraphs $G_{s,} ...., Gs$:

$$s(T) = \sum_{i=1}^{m} s(G_i)$$

- Learning: **Scoring function** $s(G_i)$ for a subgraph $G_i \in T$
- Inference: Search for maximum spanning tree scoring sequence T^* of G_s given $s(G_i)$

Graph-based Parsing (Cont.)

Learning graph-based models

- **Typical scoring function:**
  - $s(G_i) = \mathbf{w} \bullet \mathbf{f}(G_i)$ where $\mathbf{f}(G_i)$ is a high-dimensional feature vector over subgraphs and **w** is a weight vector [w_j =weight of feature $f_j(G_i)$]
- **Structured learning [McDonald *et al.* 2005a, Smith and Johnson 2007]:**
  - Learn weights that maximize the score of the correct dependency tree for every sentence in the training set
- **Problem**
  - Learning is global (trees) but features are local (subgraphs)

- **Grammar-based approach**

a. context-free dependency parsing

exploits a mapping from dependency structures to CFG structure representations and reuses parsing algorithms originally developed for CFG → chart parsing algorithms

b. constraint-based dependency parsing

- parsing viewed as a constraint satisfaction problem
- grammar defined as a set of constraints on well-formed dependency graphs
- finding a dependency graph for a sentence that satisfies all the constraints of the grammar (having the best score)

- # Grammar-based approach (Cont.)

a. context-free dependency parsing

Advantage: Well-studied parsing algorithms such as CKY, Earley's algorithm can be used for dependency parsing as well.

→ need to convert dependency grammars into efficiently parsable context-free grammars; (e.g. *bilexical CFG,* Eisner and Smith, 2005)

b. constraint-based dependency parsing

defines the problem as constraint satisfaction

- Weighted constraint dependency grammar (WCDG, Foth and Menzel, 2005)
- Transformation-based CDG

2. Dependency parsers

- Trainable parsers
  - Probabilistic dependency parser (Eisner, 1996, 2000)
  - MSTParser (McDonald, 2006)-graph-based
  - MaltParser (Nivre, 2007, 2008)-transition-based
  - *K*-best Maximum Spanning Tree Dependency Parser (Hall, 2007)
  - Vine Parser
  - ISBN Dependency Parser
- Parsers for specific languages
  - Minipar (Lin, 1998)
  - WCDG Parser (Foth *et al.*, 2005)
  - Pro3Gres (Schneider, 2004)
  - Link Grammar Parser (Lafferty *et al.*, 1992)
  - CaboCha (Kudo and Matsumoto, 2002)

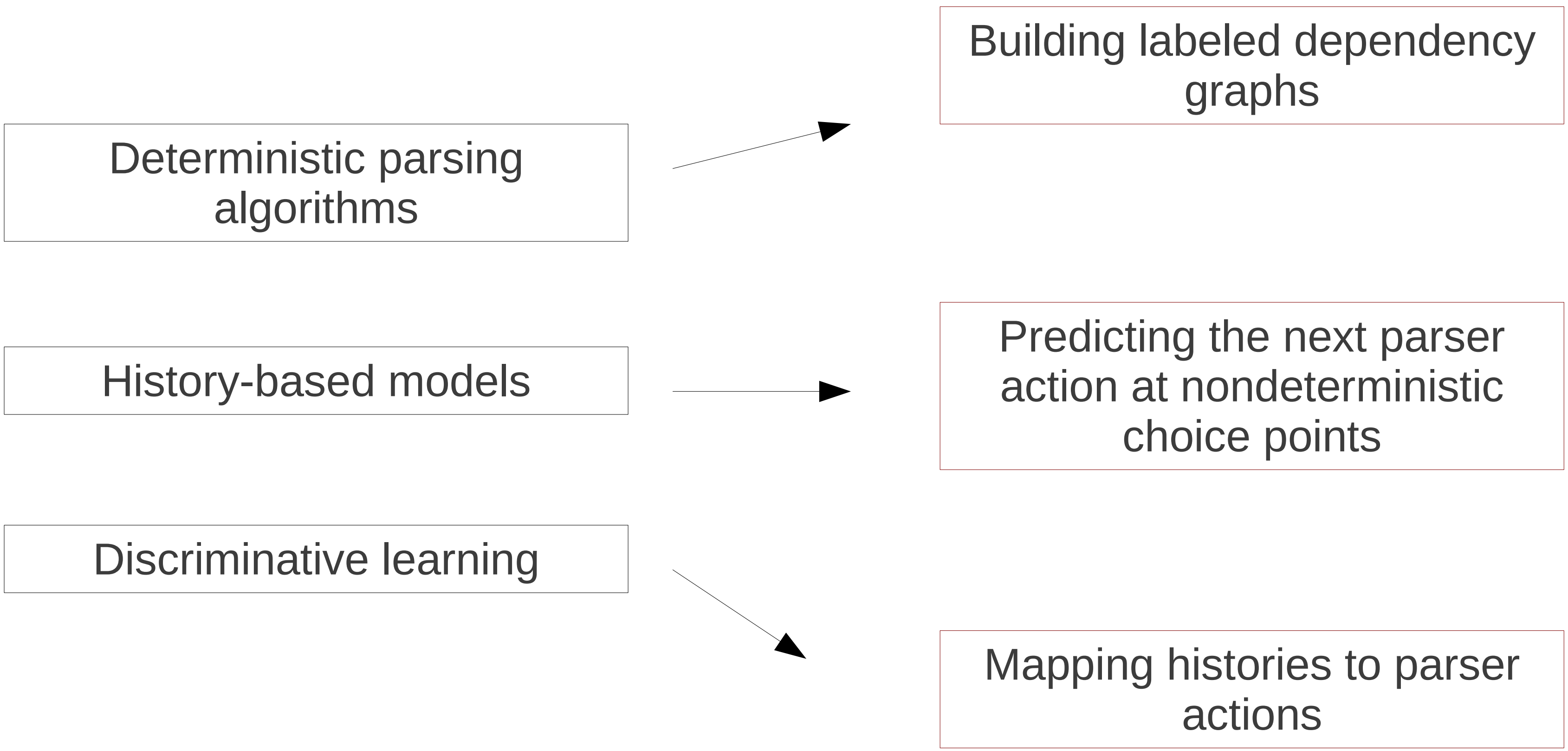
MaltParser

Data-driven dependency parsing system (Last version, 1.6.1, J. Hall, J. Nilsson and J. Nivre)

- Transition-based parsing system
- Implementation of inductive dependency parsing
- Useful for inducing a parsing model from treebank data
- Useful for parsing new data using an induced model

Useful links
http://maltparser.org

Components of system



Running system

- Input: part-of-speech tags or word forms

1	Den	_	PO	PO	DP	2	SS		_	_	
2	blir	_	V	BV	PS	0	ROOT		_	_	
3	gemensam		_	AJ	AJ	_	2	SP		_	_
4	fÃ¶r	_	PR	PR	_	2	OA		_	_	
5	alla	_	PO	PO	TP	6	DT		_	_	
6	inkomsttagare		_	N	NN	HS	4	PA		_	_
7	oavsett	_	PR	PR	_	2	AA		_	_	
8	civilstÃ¥nd		_	N	NN	SS	7	PA		_	_
9	.	_	P	IP	_	2	IP		_	_	

- Output: column containing a dependency label

MSTParser

Minimum Spanning Tree Parser (Last version, 0.2, R. McDonald *et al.*, 2005, 2006)

- Graph-based parsing system

Useful links
http://www.seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html

Running system

- Input data format:

w1 w2 ... wn

p1 p2 ... pn

l1 l2 ... ln

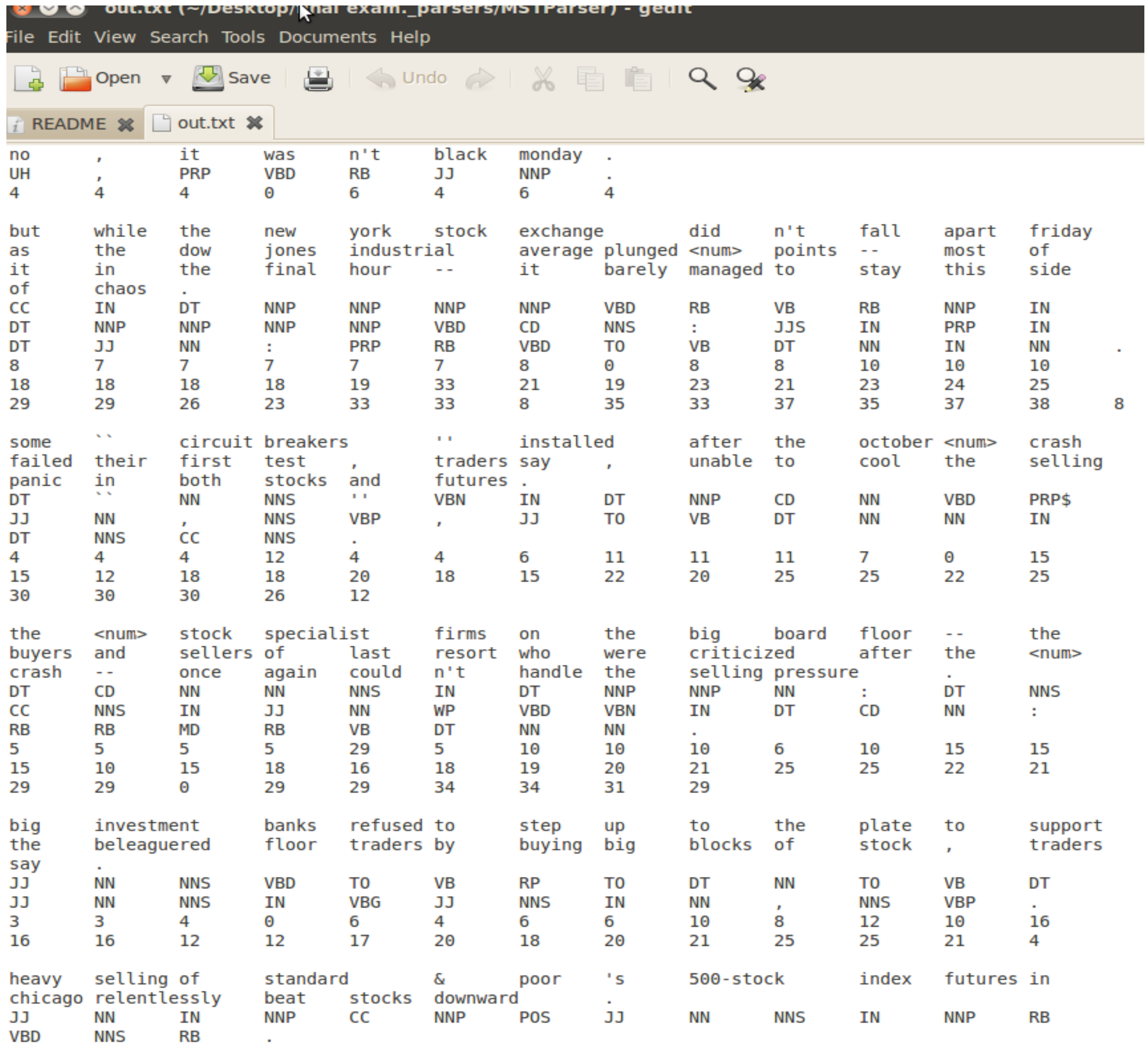
d1 d2 ... d2

Where,
- w1 ... wn are the n words of the sentence (tab deliminated)
- p1 ... pn are the POS tags for each word
- l1 ... ln are the labels of the incoming edge to each word
- d1 ... dn are integers representing the postition of each words parent

For example, the sentence *"John hit the ball"* would be:

John	hit	the	ball	
N	V	D	N	
SBJ	ROOT		MOD	OBJ
2	0	4	2	

• Output: column containing a dependency label




```
no      ,       it      was     n't     black   monday  .
UH      ,       PRP     VBD     RB      JJ      NNP     .
4       4       4       0       6       4       6       4

but     while   the     new     york    stock   exchange        did     n't     fall    apart   friday
as      the     dow     jones   industrial      average plunged <num>   points  --      most    of
it      in      the     final   hour    --      it      barely  managed to      stay    this    side
of      chaos   .
CC      IN      DT      NNP     NNP     NNP     NNP     VBD     RB      VB      RB      NNP     IN
DT      NNP     NNP     NNP     NNP     VBD     CD      NNS     :       JJS     IN      PRP     IN
DT      JJ      NN      :       PRP     RB      VBD     TO      VB      DT      NN      IN      NN      .
8       7       7       7       7       7       8       0       8       8       10      10      10
18      18      18      18      19      33      21      19      23      21      23      24      25
29      29      26      23      33      33      8       35      33      37      35      37      38      8

some    ``      circuit breakers        ''      installed       after   the     october <num>   crash
failed  their   first   test    ,       traders say     ,       unable  to      cool    the     selling
panic   in      both    stocks  and     futures .
DT      ``      NN      NNS     ''      VBN     IN      DT      NNP     CD      NN      VBD     PRP$
JJ      NN      ,       NNS     VBP     ,       JJ      TO      VB      DT      NN      NN      IN
DT      NNS     CC      NNS     .
4       4       4       12      4       4       6       11      11      11      7       0       15
15      12      18      18      20      18      15      22      20      25      25      22      25
30      30      30      26      12

the     <num>   stock   specialist      firms   on      the     big     board   floor   --      the
buyers  and     sellers of      last    resort  who     were    criticized      after   the     <num>
crash   --      once    again   could   n't     handle  the     selling pressure        .
DT      CD      NN      NN      NNS     IN      DT      NNP     NNP     NN      :       DT      NNS
CC      NNS     IN      JJ      NN      WP      VBD     VBN     IN      DT      CD      NN      :
RB      RB      MD      RB      VB      DT      NN      NN      .
5       5       5       5       29      5       10      10      10      6       10      15      15
15      10      15      18      16      18      19      20      21      25      25      22      21
29      29      0       29      29      34      34      31      29

big     investment      banks   refused to      step    up      to      the     plate   to      support
the     beleaguered     floor   traders by      buying  big     blocks  of      stock   ,       traders
say     .
JJ      NN      NNS     VBD     TO      VB      RP      TO      DT      NN      TO      VB      DT
JJ      NN      NNS     IN      VBG     JJ      NNS     IN      NN      ,       NNS     VBP     .
3       3       4       0       6       4       6       6       10      8       12      10      16
16      16      12      12      17      20      18      20      21      25      25      21      4

heavy   selling of      standard        &       poor    's      500-stock       index   futures in
chicago relentlessly    beat    stocks  downward        .
JJ      NN      IN      NNP     CC      NNP     POS     JJ      NN      NNS     IN      NNP     RB
VBD     NNS     RB      .
```

Comparing parsing accuracy
Graph-based Vs. Transition-based
MST Vs. Malt

Language	**MST**	**Malt**
Arabic	**66.91**	66.71
Bulgarian	**87.57**	87.41
Chinese	85.90	**86.92**
Czech	**80.18**	78.42
Danish	**84.79**	84.77
Dutch	**79.19**	78.59
German	**87.34**	85.82
Japanese	90.71	**91.65**
Portuguese	86.82	**87.60**
Slovene	**73.44**	70.30
Spanish	**82.25**	81.29
Swedish	82.55	**84.58**
Turkish	63.19	**65.68**
Average	**80.83**	80.75

Presented in *Current Trends in Data-Driven Dependency Parsing* by Joakim Nivre, 2009

Link Parser

Syntactic parser of English, based on the Link Grammar

(version, 4.7.4, Feb. 2011, D. Temperley, D, Sleator, J. Lafferty, 2004)

- Words as blocks with connectors + or -
- Words rules for defining the connection between the connectors
- Deep syntactic parsing system

Useful links

http://www.link.cs.cmu.edu/link/index.html
http://www.abisource.com/

- Example of a parsing in the Link Grammar:

let's test our proper sentences!

http://www.link.cs.cmu.edu/link/submit-sentence-4.html

John gives a book to Mary.

```
Press RETURN for the next linkage.
linkparser> John gives a book to Mary.
++++Time                                          0.00 seconds (0.06 total)
Found 2 linkages (2 had no P.P. violations)
  Linkage 1, cost vector = (UNUSED=0 DIS=0 AND=0 LEN=9)

    +----------------Xp----------------+
    |             +------MVp-----+     |
    |             +----Os---+    |     |
    +---Wd--+--Ss-+    +-Ds-+    +-Js+ |
    |       |     |    |    |    |   | |
LEFT-WALL John gives.v a book.n to Mary .

Press RETURN for the next linkage.
linkparser>
  Linkage 2, cost vector = (UNUSED=0 DIS=1 AND=0 LEN=7)

    +----------------Xp----------------+
    |             +----Os---+          |
    +---Wd--+--Ss-+    +-Ds-+-Mp-+-Js+ |
    |       |     |    |    |    |   | |
LEFT-WALL John gives.v a book.n to Mary .

linkparser> 
```

Some fans on Friday will be seeking to add another store-opening shirt to collections they've assembled as if they were rare baseball cards.

```
link-4.1b : parse
File  Edit  View  Scrollback  Bookmarks  Settings  Help

  +-Ss-+--Os-+
  |    |     |
John loves Mary

linkparser> Some fans on Friday will be seeking to add another store-opening shirt to collections they've asse
mbled as if they were rare baseball cards.
++++Time                                          0.06 seconds (0.06 total)
Found 276 linkages (116 had no P.P. violations)
  Linkage 1, cost vector = (UNUSED=0 DIS=3 AND=0 LEN=39)

    +------------------------------------------------------------------
    |                                                      +-----------
    |                                                      +-----------
    +------Wd-----+-------Sp-------+                       |      +----
    |       +-Dmc-+-Mp-+-ON-+      +--Ix-+-Pg*b-+--TO--+-I-+      |
    |       |     |    |    |      |     |      |      |   |      |
LEFT-WALL some fans.n on Friday will.v be.v seeking.v to add.v another


------------------Xp----------------------------------------------------------
-----MVp---------------+
--Os-------------+     |                                                  +---
------Ds---------+     |       +----------Bp---------+----MVs---+         |
      +----Ah----+     +---Jp--+----Rn---+-Sp+---PP--+       +ID+-Cs+-Spx-+
      |          |     |       |         |   |       |       |  |   |     |
store-opening shirt.n to collections.n they 've assembled.v as if they were.v


--------------------------+
                          |
--------Opt----------+    |
   +--------A--------+    |
   |        +---AN---+    |
   |        |        |    |
rare.a baseball.n cards.n .

Press RETURN for the next linkage.
```

WCDG parser

Weighted Constraint Dependency Grammar Parser

(version, 0.97-1, May, 2011, W. Menzel, N. Beuck, C. Baumgärtner)

- incremental parsing
- syntactic predictions for incomplete sentences
- Deep syntactic parsing system

Useful links

http://nats-www.informatik.uni-hamburg.de/view/CDG/ParserDemo