Dependency Parsing

L545

With thanks to Joakim Nivre and Sandra Kübler

Dependency Grammar

- Not a coherent grammatical framework: wide range of different kinds of dependency grammar
  - just as there are wide ranges of "generative syntax"
- Different core ideas than phrase structure grammar
- We will base a lot of our discussion on [Mel'čuk(1988)]

Dependency grammar is important for those interested in CL:

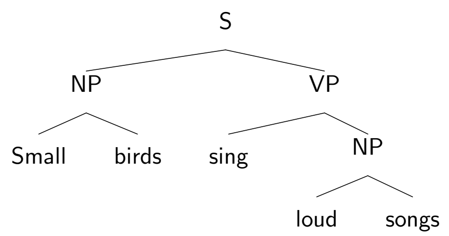
- Increasing interest in dependency-based approaches to syntactic parsing in recent years (e.g., CoNLL-X shared task, 2006)

Dependency Syntax

- The basic idea:
  - Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies.
- In the (translated) words of Lucien Tesnière [Tesnière(1959)]:
  - The sentence is an *organized whole*, the constituent elements of which are *words*. [1.2] Every word that belongs to a sentence ceases by itself to be isolated as in the dictionary. Between the word and its neighbors, the mind perceives *connections*, the totality of which forms the structure of the sentence. [1.3] The structural connections establish *dependency* relations between the words. Each connection in principle unites a *superior* term and an *inferior* term. [2.1] The superior term receives the name *governor*. The inferior term receives the name *subordinate*. Thus, in the sentence *Alfred parle* [. . .], *parle* is the governor and *Alfred* the subordinate. [2.2]

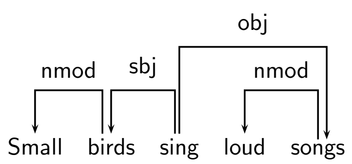
Overview: constituency

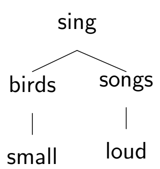
(1) Small birds sing loud songs



Overview: dependency

The corresponding dependency tree representations [Hudson(2000)]:

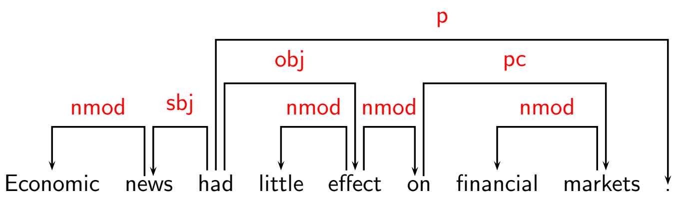




Constituency vs. Relations

- Dependency grammar is based on relationships between words, i.e., **dependency relations**
  - A → B means *A governs B* or *B depends on A* ...
  - Dependency relations can refer to syntactic properties, semantic properties, or a combination of the two
  - Relation examples: subject, object, complement, (pre-/post-)adjunct, etc.
    - Subject/Agent: *John* fished.
    - Object/Patient: Mary hit *John*.
- Phrase structure grammar is based on constituents
  - Grammatical relations are not usually seen as primitives, but as being derived from structure

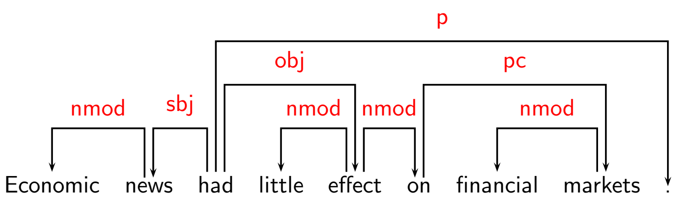
Dependency Structure

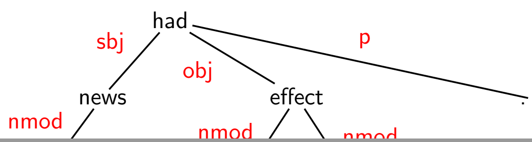


Terminology

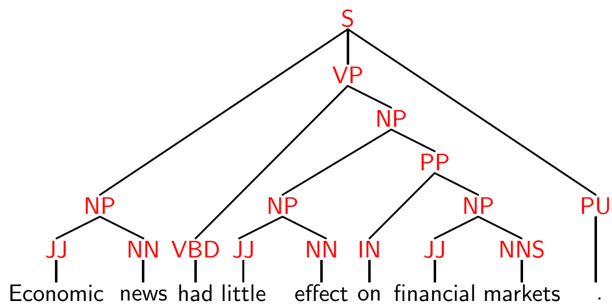
Superior	**Inferior**
Head	Dependent
Governor	Modifier
Regent	Subordinate
⋮	⋮

Notational Variants





Phrase Structure



Comparison

- Dependency structures explicitly represent
  - head-dependent relations (directed arcs),
  - functional categories (arc labels),
  - possibly some structural categories (parts-of-speech).
- Phrase structures explicitly represent
  - phrases (nonterminal nodes),
  - structural categories (nonterminal labels),
  - possibly some functional categories (grammatical functions).
- Hybrid representations may combine all elements.

Some Theoretical Frameworks

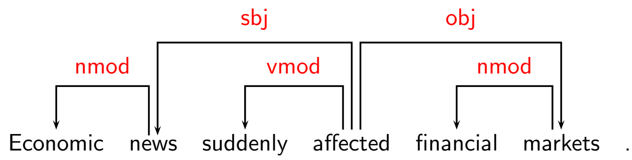
- Word Grammar (WG) [Hudson(1984), Hudson(1990)]
- Functional Generative Description (FGD) [Sgall et al.(1986)Sgall, Hajičová and Panevová]
- Dependency Unification Grammar (DUG) [Hellwig(1986), Hellwig(2003)]
- Meaning-Text Theory (MTT) [Mel'čuk(1988)]
- (Weighted) Constraint Dependency Grammar ([W]CDG) [Maruyama(1990), Harper and Helzerman(1995), Menzel and Schröder(1998), Schröder(2002)]
- Functional Dependency Grammar (FDG) [Tapanainen and Järvinen(1997), Järvinen and Tapanainen(1998)]
- Topological/Extensible Dependency Grammar ([T/X]DG) [Duchier and Debusmann(2001), Debusmann et al.(2004)Debusmann, Duchier and Kruijff]

Some Theoretical Issues

- Dependency structure sufficient as well as necessary?
- Mono-stratal or multi-stratal syntactic representations?
- What is the nature of lexical elements (nodes)?
  - Morphemes?
  - Word forms?
  - Multi-word units?
- What is the nature of dependency types (arc labels)?
  - Grammatical functions?
  - Semantic roles?
- What are the criteria for identifying heads and dependents?
- What are the formal properties of dependency structures?

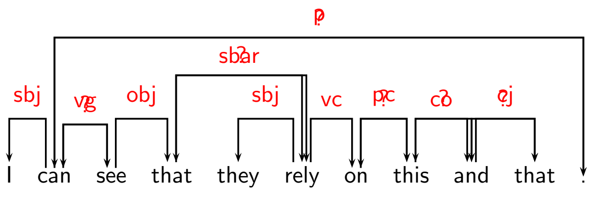
Some Clear Cases

Construction	Head	Dependent
Exocentric	Verb	Subject (sbj)
	Verb	Object (obj)
Endocentric	Verb	Adverbial (vmod)
	Noun	Attribute (nmod)



Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation



Dependency Graphs

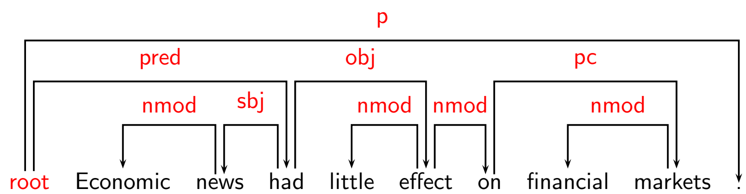
- A dependency structure can be defined as a directed graph G, consisting of
  - a set V of nodes,
  - a set E of arcs (edges),
  - a linear precedence order $<$ on V.
- Labeled graphs:
  - Nodes in V are labeled with word forms (and annotation).
  - Arcs in E are labeled with dependency types.
- Notational conventions ($i, j \in V$):
  - $i \to j \;\equiv\; (i, j) \in E$
  - $i \to^* j \;\equiv\; i = j \lor \exists k : i \to k, k \to^* j$

Formal Conditions on Dependency Graphs

- G is (weakly) connected:
  - For every node i there is a node j such that $i \to j$ or $j \to i$.
- G is acyclic:
  - If $i \to j$ then not $j \to^* i$.
- G obeys the single-head constraint:
  - If $i \to j$, then not $k \to j$, for any $k \neq i$.
- G is projective:
  - If $i \to j$ then $i \to^* k$, for any k such that $i < k < j$ or $j < k < i$.

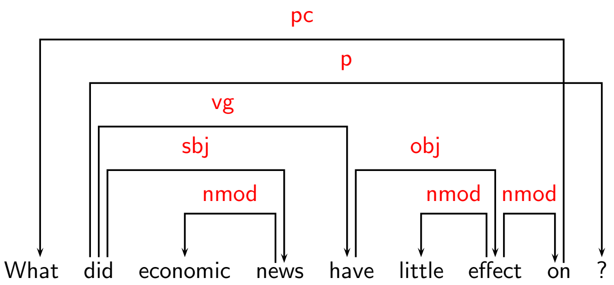
Connectedness, Acyclicity and Single-Head

- Intuitions:
  - Syntactic structure is complete (Connectedness).
  - Syntactic structure is hierarchical (Acyclicity).
  - Every word has at most one syntactic head (Single-Head).
- Connectedness can be enforced by adding a special root node.



Projectivity

- Most theoretical frameworks do not assume projectivity.
- Non-projective structures are needed to account for
  - long-distance dependencies,
  - free word order.



Where we're going

- Dependency parsing:
  - Input: Sentence $x = w_1, \ldots w_n$
  - Output: Dependency graph G
- Focus today:
  - Computational methods for dependency parsing
  - Resources for dependency parsing (parsers, treebanks)

Parsing Methods

- Three main traditions:
  - Deterministic parsing (specifically: **Transition-based parsing**)
  - Dynamic programming (specifically: **Graph-based parsing**)
  - Constraint satisfaction (not covered today)
- Special issue:
  - Non-projective dependency parsing

Deterministic Parsing

- Basic idea:
  - Derive a single syntactic representation (dependency graph) through a deterministic sequence of elementary parsing actions
  - Sometimes combined with backtracking or repair
- Motivation:
  - Psycholinguistic modeling
  - Efficiency
  - Simplicity

Covington's Incremental Algorithm

- Deterministic incremental parsing in $O(n^2)$ time by trying to link each new word to each preceding one [Covington(2001)]:

```
PARSE(x = (w_1, ..., w_n))
  for i = 1 up to n
    for j = i − 1 down to 1
      LINK(w_i, w_j)
```

$$\textsc{Link}(w_i, w_j) = \begin{cases} E \leftarrow E \cup (i,j) & \text{if } w_j \text{ is a dependent of } w_i \\ E \leftarrow E \cup (j,i) & \text{if } w_i \text{ is a dependent of } w_j \\ E \leftarrow E & \text{otherwise} \end{cases}$$

- Different conditions, such as Single-Head and Projectivity, can be incorporated into the LINK operation.

Shift-Reduce Type Algorithms

Transition-based parsing

- Data structures:
  - Stack $[\ldots, w_i]_S$ of partially processed tokens
  - Queue $[w_j, \ldots]_Q$ of remaining input tokens
- Parsing actions built from atomic actions:
  - Adding arcs ($w_i \rightarrow w_j$, $w_i \leftarrow w_j$)
  - Stack and queue operations
- Left-to-right parsing in $O(n)$ time
- Restricted to projective dependency graphs

Yamada's Algorithm

- Three parsing actions:

$$\text{Shift}\quad \frac{[\ldots]_S \quad [w_i,\ldots]_Q}{[\ldots,w_i]_S \quad [\ldots]_Q}$$

$$\text{Left}\quad \frac{[\ldots,w_i,w_j]_S \quad [\ldots]_Q}{[\ldots,w_i]_S \quad [\ldots]_Q \quad w_i \rightarrow w_j}$$

$$\text{Right}\quad \frac{[\ldots,w_i,w_j]_S \quad [\ldots]_Q}{[\ldots,w_j]_S \quad [\ldots]_Q \quad w_i \leftarrow w_j}$$

- Algorithm variants:
  - Originally developed for Japanese (strictly head-final) with only the Shift and Right actions [Kudo and Matsumoto(2002)]
  - Adapted for English (with mixed headedness) by adding the Left action [Yamada and Matsumoto(2003)]
  - Multiple passes over the input give time complexity $O(n^2)$

Nivre's Algorithm

- Four parsing actions:

$$\text{Shift}\quad \frac{[\ldots]_S \quad [w_i,\ldots]_Q}{[\ldots,w_i]_S \quad [\ldots]_Q}$$

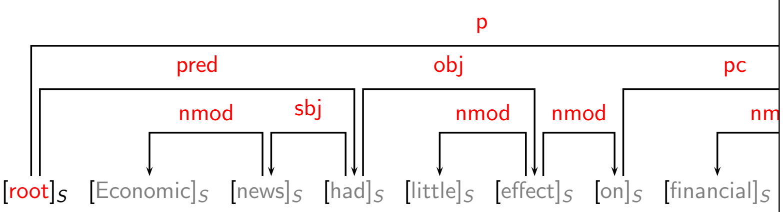
$$\text{Reduce}\quad \frac{[\ldots,w_i]_S \quad [\ldots]_Q \quad \exists w_k : w_k \rightarrow w_i}{[\ldots]_S \quad [\ldots]_Q}$$

$$\text{Left-Arc}_r\quad \frac{[\ldots,w_i]_S \quad [w_j,\ldots]_Q \quad \neg\exists w_k : w_k \rightarrow w_i}{[\ldots]_S \quad [w_j,\ldots]_Q \quad w_i \stackrel{r}{\leftarrow} w_j}$$

$$\text{Right-Arc}_r\quad \frac{[\ldots,w_i]_S \quad [w_j,\ldots]_Q \quad \neg\exists w_k : w_k \rightarrow w_j}{[\ldots,w_i,w_j]_S \quad [\ldots]_Q \quad w_i \stackrel{r}{\rightarrow} w_j}$$

- Characteristics:
  - Integrated labeled dependency parsing
  - Arc-eager processing of right-dependents
  - Single pass over the input gives time complexity $O(n)$

Example



Shift Left-Arc$_{nmod}$ Shift Left-Arc$_{sbj}$ Right-Arc$_{pred}$ Shift Left-Arc$_{nmod}$ Right-Arc$_{obj}$ Right-Arc$_{nmod}$ Shift Left-Arc$_{nmod}$ Right-Arc$_{pc}$ Reduce Reduce Reduce Reduce Right-Arc$_{p}$

Classifier-Based Parsing

- Data-driven deterministic parsing:
  - Deterministic parsing requires an oracle.
  - An oracle can be approximated by a classifier.
  - A classifier can be trained using treebank data.
- Learning methods:
  - Support vector machines (SVM) [Kudo and Matsumoto(2002), Yamada and Matsumoto(2003), Isozaki et al.(2004)Isozaki, Kazawa and Hirao, Cheng et al.(2004)Cheng, Asahara and Matsumoto, Nivre et al.(2006)Nivre, Hall, Nilsson, Eryiğit and Marinov]
  - Memory-based learning (MBL) [Nivre et al.(2004)Nivre, Hall and Nilsson, Nivre and Scholz(2004)]
  - Maximum entropy modeling (MaxEnt) [Cheng et al.(2005)Cheng, Asahara and Matsumoto]

Feature Models

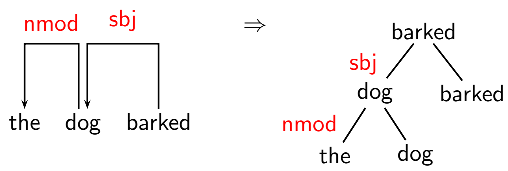
- Learning problem:
  - Approximate a function from parser states, represented by feature vectors to parser actions, given a training set of gold standard derivations.
- Typical features:
  - Tokens:
    - Target tokens
    - Linear context (neighbors in S and Q)
    - Structural context (parents, children, siblings in G)
  - Attributes:
    - Word form (and lemma)
    - Part-of-speech (and morpho-syntactic features)
    - Dependency type (if labeled)
    - Distance (between target tokens)

Comparing Algorithms

- Parsing algorithm:
  - Nivre's algorithm gives higher accuracy than Yamada's algorithm for parsing the Chinese CKIP treebank [Cheng et al.(2004)Cheng, Asahara and Matsumoto].
- Learning algorithm:
  - SVM gives higher accuracy than MaxEnt for parsing the Chinese CKIP treebank [Cheng et al.(2004)Cheng, Asahara and Matsumoto].
  - SVM gives higher accuracy than MBL with lexicalized feature models for three languages [Hall et al.(2006)Hall, Nivre and Nilsson]:
    - Chinese (Penn)
    - English (Penn)
    - Swedish (Talbanken)

Dynamic Programming

- Basic idea: Treat dependencies as constituents.
- Use, e.g., CYK parser (with minor modifications).
- Dependencies as constituents:



Dependency Chart Parsing

- Grammar is regarded as context-free, in which each node is lexicalized.
- Chart entries are subtrees, i.e., words with all their left and right dependents.
- Problem: Different entries for different subtrees spanning a sequence of words with different heads.
- Time requirement: $O(n^5)$.

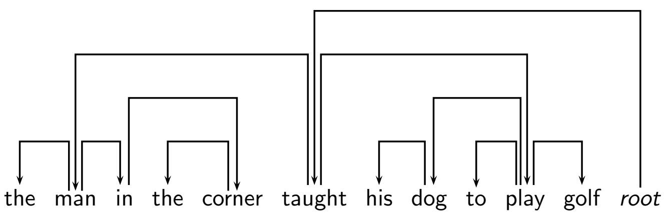
Dynamic Programming Approaches

- Original version: [Hays(1964)]
- Link Grammar: [Sleator and Temperley(1991)]
- Earley-style parser with left-corner filtering: [Lombardo and Lesmo(1996)]
- Bilexical grammar: [Eisner(1996a), Eisner(1996b), Eisner(2000)]
- Bilexical grammar with discriminative estimation methods: [McDonald et al.(2005a)McDonald, Crammer and Pereira, McDonald et al.(2005b)McDonald, Pereira, Ribarov and Hajič]

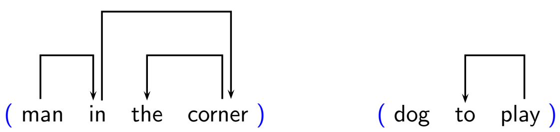
Eisner's Bilexical Algorithm

- Two novel aspects:
  - Modified parsing algorithm
  - Probabilistic dependency parsing
- Time requirement: $O(n^3)$.
- Modification: Instead of storing subtrees, store spans.
- Def. span: Substring such that no interior word links to any word outside the span.
- Underlying idea: In a span, only the endwords are active, i.e. still need a head.
- One or both of the endwords can be active.

Example

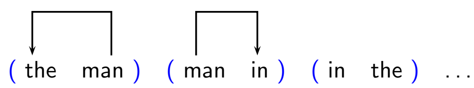


Spans:

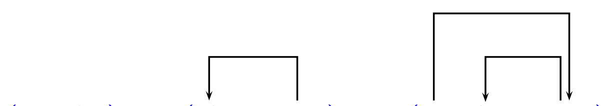


Assembly of Correct Parse

Start by combining adjacent words to minimal spans:



Combine spans which overlap in one word; this word must be governed by a word in the left or right span.



Eisner's Probability Models

- Model A: Bigram lexical affinities
  - First generates a trigram Markov model for POS tagging.
  - Decides for each word pair whether they have a dependency.
  - Model is leaky because it does not control for crossing dependencies, multiple heads, ...
- Model B: Selectional preferences
  - First generates a trigram Markov model for POS tagging.
  - Each word chooses a subcat/supercat frame.
  - Selects an analysis that satisfies all frames if possible.
  - Model is also leaky because last step may fail.
- Model C: Recursive Generation
  - Each word generates its actual dependents.
  - Two Markov chains:
    - Left dependents
    - Right dependents
  - Model is not leaky.

Eisner's Model C

$$Pr(words, tags, links) = \prod_{1 \le i \le n} \left(\prod_{c} Pr(tword(dep_c(i)) \mid tag(dep_{c-1}(i)), tword(i)) \right)$$

$c = -(1 + \#left - deps(i)) \ldots 1 + \#right - deps(i),\ c \neq 0$

or: $dep_{c+i}(i)$ if $c < 0$

Eisner's Results

- 25 000 Wall Street Journal sentences
- Baseline: most frequent tag chosen for a word, each word chooses a head with most common distance
- Model X: trigram tagging, no dependencies
- For comparison: state-of-the-art constituent parsing, Charniak: 92.2 F-measure

Model	Non-punct	Tagging
Baseline	41.9	76.1
Model X	–	93.1
Model A	too slow	
Model B	83.8	92.8
Model C	86.9	92.0

Maximum Spanning Trees

[McDonald et al.(2005a)McDonald, Crammer and Pereira, McDonald et al.(2005b)McDonald, Pereira, Ribarov and Hajič]

Graph-based parsing

- Score of a dependency tree = sum of scores of dependencies
- Scores are independent of other dependencies.
- If scores are available, parsing can be formulated as maximum spanning tree problem.
- Two cases:
  - Projective: Use Eisner's parsing algorithm.
  - Non-projective: Use Chu-Liu-Edmonds algorithm for finding the maximum spanning tree in a directed graph [Chu and Liu(1965), Edmonds(1967)].
- Use online learning for determining weight vector **w**: large-margin multi-class classification (MIRA)

Maximum Spanning Trees (2)

- Complexity:
  - Projective (Eisner): $O(n^3)$
  - Non-projective (CLE): $O(n^2)$

$$score(sent, deps) = \sum_{(i,j) \in deps} score(i,j) = \sum_{(i,j) \in deps} \mathbf{w} \cdot f(i,j)$$

Online Learning

Training data: $\mathcal{T} = (sent_t, deps_t)_{t=1}^{\mathcal{T}}$

1. $\mathbf{w} = 0$; $\mathbf{v} = 0$; $i = 0$;
2. for $n : 1..\mathcal{N}$
3. for $t : 1..\mathcal{T}$
4. $\mathbf{w}^{(i+1)}$ = update $\mathbf{w}^{(i)}$ according to $(sent_t, deps_t)$
5. $\mathbf{v} = \mathbf{v} + \mathbf{w}^{(i+1)}$
6. $i = i + 1$
7. $\mathbf{w} = \mathbf{v}/(\mathcal{N} \cdot \mathcal{T})$

MIRA

MIRA weight update:

min $||\mathbf{w}^{(i+1)} - \mathbf{w}^{(i)}||$ so that

$$score(sent_t, deps_t) - score(sent_t, deps') \geq L(deps_t, deps')$$

$$\forall deps' \in dt(sent_t)$$

- $L(deps, deps')$: loss function
- $dt(sent)$: possible dependency parses for sentence

Results by McDonald et al. (2005a, 2005b)

- Unlabeled accuracy per word (W) and per sentence (S)

	English		Czech	
Parser	W	S	W	S
k-best MIRA Eisner	90.9	37.5	83.3	31.3
best MIRA CLE	90.2	33.2	84.1	32.2
factored MIRA CLE	90.2	32.2	84.4	32.3

- New development (EACL 2006):
  - Scores of dependencies are not independent any more
  - Better results

Evaluation on English

- Evaluation:
  - Penn Treebank (WSJ) converted to dependency graphs
  - Unlabeled accuracy per word (W) and per sentence (S)
    - Deterministic classifier-based parsers [Yamada and Matsumoto(2003), Isozaki et al.(2004)Isozaki, Kazawa and Hirao]
    - Spanning tree parsers with online training [McDonald et al.(2005a)McDonald, Crammer and Pereira, McDonald and Pereira(2006)]
    - Collins and Charniak parsers with same conversion

Parser	W	S
Charniak	92.2	45.2
Collins	91.7	43.3
McDonald and Pereira	91.5	42.1
Isozaki et al.	91.4	40.7
McDonald et al.	91.0	37.5
Yamada and Matsumoto	90.4	38.4

Parsing Methods

- Three main traditions:
  - Deterministic parsing (specifically: **Transition-based parsing**)
  - Dynamic programming (specifically: **Graph-based parsing**)
  - Deterministic parsing
- Special issue:
  - Non-projective dependency parsing

Non-Projective Dependency Parsing

- Many parsing algorithms are restricted to projective dependency graphs.
- Is this a problem?
- Statistics from CoNLL-X Shared Task [Buchholz and Marsi(2006)]
  - NPD = Non-projective dependencies
  - NPS = Non-projective sentences

Language	%NPD	%NPS
Dutch	5.4	36.4
German	2.3	27.8
Czech	1.9	23.2
Slovene	1.9	22.2
Portuguese	1.3	18.9
Danish	1.0	15.6

Two Main Approaches

- Algorithms for non-projective dependency parsing:
  - Constraint satisfaction methods [Tapanainen and Järvinen(1997), Duchier and Debusmann(2001), Foth et al.(2004)Foth, Daum and Menzel]
  - McDonald's spanning tree algorithm [McDonald et al.(2005b)McDonald, Pereira, Ribarov and Hajič]
  - Covington's algorithm [Nivre(2006)]
- Post-processing of projective dependency graphs:
  - Pseudo-projective parsing [Nivre and Nilsson(2005)]
  - Corrective modeling [Hall and Novák(2005)]
  - Approximate non-projective parsing [McDonald and Pereira(2006)]

Non-Projective Parsing Algorithms

- Complexity considerations:
  - Projective (Proj)
  - Non-projective (NonP)

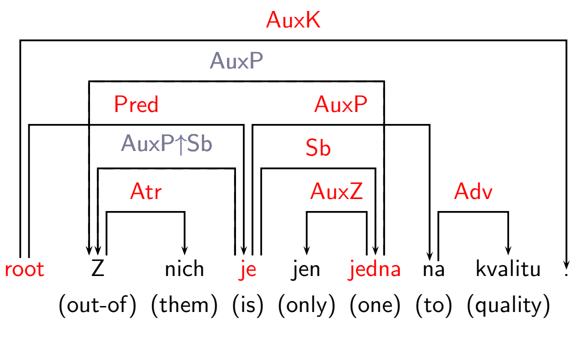
Problem/Algorithm	Proj	NonP
Complete grammar parsing [Gaifman(1965), Neuhaus and Bröker(1997)]	P	NP hard
Deterministic parsing [Nivre(2003), Covington(2001)]	$O(n)$	$O(n^2)$
First order spanning tree [McDonald et al.(2005b)McDonald, Pereira, Ribarov and Hajič]	$O(n^3)$	$O(n^2)$
Nth order spanning tree ($N > 1$) [McDonald and Pereira(2006)]	P	NP hard

Post-Processing

- Two-step approach:
  1. Derive the best projective approximation of the correct (possibly) non-projective dependency graph.
  2. Improve the approximation by replacing projective arcs by (possibly) non-projective arcs.
- Rationale:
  - Most "naturally occurring" dependency graphs are primarily projective, with only a few non-projective arcs.
- Approaches:
  - Pseudo-projective parsing [Nivre and Nilsson(2005)]
  - Corrective modeling [Hall and Novák(2005)]
  - Approximate non-projective parsing [McDonald and Pereira(2006)]

Pseudo-Projective Parsing

- Projectivize training data:
  - Projective head nearest permissible ancestor of real head
  - Arc label extended with dependency type of real head



Evaluation on Czech

- Evaluation:
  - Prague Dependency Treebank (PDT)
  - Unlabeled accuracy per word (W) and per sentence (S)
    - Non-projective spanning tree parsing [McDonald et al.(2005b)McDonald, Pereira, Ribarov and Hajič]
    - Corrective modeling on top of the Charniak parser [Hall and Novák(2005)]
    - Approximate non-projective parsing on top of a second-order projective spanning tree parser [McDonald and Pereira(2006)]
    - Pseudo-projective parsing on top of a deterministic classifier-based parser [Nilsson et al.(2006)Nilsson, Nivre and Hall]

Parser	W	S
McDonald and Pereira	85.2	35.9
Hall and Novák	85.1	—
Nilsson et al.	84.6	37.7
McDonald et al.	84.4	32.3
Charniak	84.4	–

Multilingual Parsing

- CoNLL-X Shared Task: 12 (13) languages
- Organizers: Sabine Buchholz, Erwin Marsi, Yuval Krymolowski, Amit Dubey
- Main evaluation metric: Labeled accuracy per word
- Top scores ranging from 91.65 (Japanese) to 65.68 (Turkish)
- Top systems (over all languages):
  - Approximate second-order non-projective spanning tree parsing with online learning (MIRA) [McDonald et al.(2006)McDonald, Lerman and Pereira]
  - Labeled deterministic pseudo-projective parsing with support vector machines [Nivre et al.(2006)Nivre, Hall, Nilsson, Eryiğit and Marinov]

Practical Issues

- Where to get the software?
  - Dependency parsers
  - Conversion programs for constituent-based treebanks
- Where to get the data?
  - Dependency treebanks
  - Treebanks that can be converted into dependency representation

Parsers

- Trainable parsers
- Parsers with manually written grammars

- Concentrate on freely available parsers

Trainable Parsers

- Jason Eisner's **probabilistic dependency parser**
  - Based on bilexical grammar
  - Contact Jason Eisner: jason@cs.jhu.edu
  - Written in LISP
- Ryan McDonald's **MSTParser**
  - Based on the algorithms of [McDonald et al.(2005a)McDonald, Crammer and Pereira, McDonald et al.(2005b)McDonald, Pereira, Ribarov and Hajič]
  - URL: `http://www.seas.upenn.edu/~ryantm/software/MSTParser/`
  - Written in JAVA

Trainable Parsers (2)

- Joakim Nivre's **MaltParser**
  - Inductive dependency parser with memory-based learning and SVMs
  - URL: `http://w3.msi.vxu.se/~nivre/research/MaltParser.html`
  - Executable versions are available for Solaris, Linux, Windows, and MacOS (open source version planned for fall 2006)

Parsers for Specific Languages

- Dekang Lin's **Minipar**
  - Principle-based parser
  - Grammar for English
  - URL: `http://www.cs.ualberta.ca/~lindek/minipar.htm`
  - Executable versions for Linux, Solaris, and Windows
- Wolfgang Menzel's **CDG Parser**:
  - Weighted constraint dependency parser
  - Grammar for German, (English under construction)
  - Online demo: `http://nats-www.informatik.uni-hamburg.de/Papa/ParserDemo`
  - Download: `http://nats-www.informatik.uni-hamburg.de/download`

Parsers for Specific Languages (2)

- Taku Kudo's **CaboCha**
  - Based on algorithms of [Kudo and Matsumoto(2002)], uses SVMs
  - URL: `http://www.chasen.org/~taku/software/cabocha/`
  - Web page in Japanese
- Gerold Schneider's **Pro3Gres**
  - Probability-based dependency parser
  - Grammar for English
  - URL: `http://www.ifi.unizh.ch/CL/gschneid/parser/`
  - Written in PROLOG
- Daniel Sleator's & Davy Temperley's **Link Grammar Parser**
  - Undirected links between words
  - Grammar for English
  - URL: `http://www.link.cs.cmu.edu/link/`

Treebanks

- Genuine dependency treebanks
- Treebanks for which conversions to dependencies exist

- See also CoNLL-X Shared Task
  URL: `http://nextens.uvt.nl/~conll/`

- Conversion strategy from constituents to dependencies

Dependency Treebanks

- Arabic: Prague Arabic Dependency Treebank
- Czech: Prague Dependency Treebank
- Danish: Danish Dependency Treebank
- Portuguese: Bosque: Floresta sintá(c)tica
- Slovene: Slovene Dependency Treebank
- Turkish: METU-Sabanci Turkish Treebank

Dependency Treebanks (2)

- Prague Arabic Dependency Treebank
  - ca. 100 000 words
  - Available from LDC, license fee (CoNLL-X shared task data, catalogue number LDC2006E01)
  - URL: `http://ufal.mff.cuni.cz/padt/`
- Prague Dependency Treebank
  - 1.5 million words
  - 3 layers of annotation: morphological, syntactical, tectogrammatical
  - Available from LDC, license fee (CoNLL-X shared task data, catalogue number LDC2006E02)
  - URL: `http://ufal.mff.cuni.cz/pdt2.0/`

Dependency Treebanks (3)

- Danish Dependency Treebank
  - ca. 5 500 trees
  - Annotation based on Discontinuous Grammar [Kromann(2005)]
  - Freely downloadable
  - URL: `http://www.id.cbs.dk/~mtk/treebank/`
- Bosque, Floresta sintá(c)tica
  - ca. 10 000 trees
  - Freely downloadable
  - URL: `http://acdc.linguateca.pt/treebank/info_floresta_English.html`

Dependency Treebanks (4)

- Slovene Dependency Treebank
  - ca. 30 000 words
  - Freely downloadable
  - URL: `http://nl.ijs.si/sdt/`
- METU-Sabanci Turkish Treebank
  - ca. 7 000 trees
  - Freely available, license agreement
  - URL: `http://www.ii.metu.edu.tr/~corpus/treebank.html`

Constituent Treebanks

- English: Penn Treebank
- Bulgarian: BulTreebank
- Chinese: Penn Chinese Treebank, Sinica Treebank
- Dutch: Alpino Treebank for Dutch
- German: TIGER/NEGRA, TüBa-D/Z
- Japanese: TüBa-J/S
- Spanish: Cast3LB
- Swedish: Talbanken05

Constituent Treebanks (2)

- Penn Treebank
  - ca. 1 million words
  - Available from LDC, license fee
  - URL: `http://www.cis.upenn.edu/~treebank/home.html`
  - Dependency conversion rules, available from e.g. [Collins(1999)]
  - For conversion with arc labels: Penn2Malt: `http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html`
- BulTreebank
  - ca. 14 000 sentences
  - URL: `http://www.bultreebank.org/`
  - Dependency version available from Kiril Simov (`kivs@bultreebank.org`)

Constituent Treebanks (3)

- Penn Chinese Treebank
  - ca. 4 000 sentences
  - Available from LDC, license fee
  - URL: `http://www.cis.upenn.edu/~chinese/ctb.html`
  - For conversion with arc labels: Penn2Malt: `http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html`
- Sinica Treebank
  - ca. 61 000 sentences
  - Available Academia Sinica, license fee
  - URL: `http://godel.iis.sinica.edu.tw/CKIP/engversion/treebank.htm`
  - Dependency version available from Academia Sinica

Constituent Treebanks (4)

- Alpino Treebank for Dutch
  - ca. 150 000 words
  - Freely downloadable
  - URL: `http://www.let.rug.nl/vannoord/trees/`
  - Dependency version downloadable at `http://nextens.uvt.nl/~conll/free_data.html`
- TIGER/NEGRA
  - ca. 50 000/20 000 sentences
  - Freely available, license agreement
  - TIGER URL: `http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/`
    NEGRA URL: `http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/`
  - Dependency version of TIGER is included in release

Constituent Treebanks (5)

- TüBa-D/Z
  - ca. 22 000 sentences
  - Freely available, license agreement
  - URL: `http://www.sfs.uni-tuebingen.de/en_tuebadz.shtml`
  - Dependency version available from SfS Tübingen
- TüBa-J/S
  - Dialog data
  - ca. 18 000 sentences
  - Freely available, license agreement
  - Dependency version available from SfS Tübingen
  - URL: `http://www.sfs.uni-tuebingen.de/en_tuebajs.shtml` (under construction)

Constituent Treebanks (6)

- Cast3LB
  - ca. 18 000 sentences
  - URL: `http://www.dlsi.ua.es/projectes/3lb/index_en.html`
  - Dependency version available from Toni Martí (amarti@ub.edu)
- Talbanken05
  - ca. 300 000 words
  - Freely downloadable
  - URL: `http://w3.msi.vxu.se/~nivre/research/Talbanken05.html`
  - Dependency version also available

- Buchholz, Sabine and Erwin Marsi (2006). CoNLL-X Shared Task on Multilingual Dependency Parsing.
  In *Proceedings of the Tenth Conference on Computational Natural Language Learning*.
- Cheng, Yuchang, Masayuki Asahara and Yuji Matsumoto (2004). Determinstic Dependency Structure Analyzer for Chinese.
  In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP)*. pp. 500–508.
- Cheng, Yuchang, Masayuki Asahara and Yuji Matsumoto (2005). Machine Learning-Based Dependency Analyzer for Chinese.
  In *Proceedings of International Conference on Chinese Computing (ICCC)*. pp. ?–?
- Chu, Y. J. and T. J. Liu (1965). On the Shortest Arborescence of a Directed Graph.
  Science Sinica 14, 1396–1400.
- Collins, Michael (1999). Head-Driven Statistical Models for Natural Language Parsing.
  Ph.D. thesis, University of Pennsylvania.
- Covington, Michael A. (2001). A Fundamental Algorithm for Dependency Parsing.
  In *Proceedings of the 39th Annual ACM Southeast Conference*. pp. 95–102.

- Debusmann, Ralph, Denys Duchier and Geert-Jan M. Kruijff (2004). Extensible Dependency Grammar: A New Methodology.
  In *Proceedings of the Workshop on Recent Advances in Dependency Grammar*. pp. 78–85.
- Dubey, Amit (2005). What to Do when Lexicalization Fails: Parsing German with Suffix Analysis and Smoothing.
  In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Ann Arbor, MI.
- Duchier, Denys (1999). Axiomatizing Dependency Parsing Using Set Constraints.
  In *Proceedings of the Sixth Meeting on Mathematics of Language*. pp. 115–126.
- Duchier, Denys (2003). Configuration of Labeled Trees under Lexicalized Constraints and Principles.
  Research on Language and Computation 1, 307–336.
- Duchier, Denys and Ralph Debusmann (2001). Topological Dependency Trees: A Constraint-based Account of Linear Precedence.
  In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*. pp. 180–187.
- Edmonds, J. (1967). Optimum Branchings.
  Journal of Research of the National Bureau of Standards 71B, 233–240.

- Eisner, Jason M. (1996a). *An empirical comparison of probability models for dependency grammar*.
  Tech. Rep. IRCS-96-11, Institute for Research in Cognitive Science, University of Pennsylvania.
- Eisner, Jason M. (1996b). Three new probabilistic models for dependency parsing: An exploration.
  In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*. pp. 340–345.
- Eisner, Jason M. (2000). Bilexical grammars and their cubic-time parsing algorithms.
  In Harry Bunt and Anton Nijholt (eds.), *Advances in Probabilistic and Other Parsing Technologies*, Kluwer, pp. 29–62.
- Foth, Kilian, Michael Daum and Wolfgang Menzel (2004). A Broad-Coverage Parser for German Based on Defeasible Constraints.
  In *Proceedings of KONVENS 2004*. pp. 45–52.
- Foth, Kilian, Ingo Schröder and Wolfgang Menzel (2000). A Transformation-based parsing technique with anytime properties.
  In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT)*. pp. 89–100.
- Gaifman, Haim (1965). Dependency systems and phrase-structure systems.

Information and Control 8, 304–337.
- Hall, Johan, Joakim Nivre and Jens Nilsson (2006). Discriminative Classifiers for Deterministic Dependency Parsing.
  In *Proceedings of COLING-ACL*.
- Hall, Keith and Vaclav Novák (2005). Corrective modeling for non-projective dependency parsing.
  In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*. pp. 42–52.
- Harper, Mary P. and R. A. Helzerman (1995). Extensions to constraint dependency parsing for spoken language processing.
  Computer Speech and Language 9, 187–234.
- Hays, David G. (1964). Dependency Theory: A Formalism and Some Observations.
  Language 40, 511–525.
- Hellwig, Peter (1986). Dependency Unification Grammar.
  In *Proceedings of the 11th International Conference on Computational Linguistics (COLING)*. pp. 195–198.
- Hellwig, Peter (2003). Dependency Unification Grammar.
  In Vilmos Agel, Ludwig M. Eichinger, Hans-Werner Eroms, Peter Hellwig, Hans Jürgen Heringer and Hening Lobin (eds.), *Dependency and Valency*, Walter de Gruyter, pp. 593–635.

- Hudson, Richard A. (1984). *Word Grammar*.
  Blackwell.
- Hudson, Richard A. (1990). *English Word Grammar*.
  Blackwell.
- Hudson, Richard A. (2000). Dependency Grammar Course Notes.
  `http://www.cs.bham.ac.uk/research/conferences/esslli/notes/hudson.\html`.
- Isozaki, Hideki, Hideto Kazawa and Tsutomu Hirao (2004). A Deterministic Word Dependency Analyzer Enhanced with Preference Learning.
  In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*. pp. 275–281.
- Järvinen, Timo and Pasi Tapanainen (1998). Towards an Implementable Dependency Grammar.
  In Sylvain Kahane and Alain Polguère (eds.), *Proceedings of the Workshop on Processing of Dependency-Based Grammars*. pp. 1–10.
- Karlsson, Fred (1990). Constraint grammar as a framework for parsing running text.
  In Hans Karlgren (ed.), *Papers presented to the 13th International Conference on Computational Linguistics (COLING)*. pp. 168–173.
- Karlsson, Fred, Atro Voutilainen, Juha Heikkilä and Arto Anttila (eds.) (1995). *Constraint Grammar: A language-independent system for parsing unrestricted text*.

Mouton de Gruyter.
- Kromann, Matthias Trautner (2005). *Discontinuous Grammar: A Dependency-Based Model of Human Parsing and Language Learning*.
  Doctoral Dissertation, Copenhagen Business School.
- Kübler, Sandra, Erhard W. Hinrichs and Wolfgang Maier (2006). Is it Really that Difficult to Parse German?
  In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP 2006*. Sydney, Australia.
- Kudo, Taku and Yuji Matsumoto (2002). Japanese Dependency Analysis Using Cascaded Chunking.
  In *Proceedings of the Sixth Workshop on Computational Language Learning (CoNLL)*. pp. 63–69.
- Lin, Dekang (1995). A Dependency-Based Method for Evaluating Broad-Coverage Parsers.
  In *Proceedings of IJCAI-95*. pp. 1420–1425.
- Lin, Dekang (1998). A Dependency-Based Method for Evaluating Broad-Coverage Parsers.
  Natural Language Engineering 4, 97–114.
- Lombardo, Vincenzio and Leonardo Lesmo (1996). An Earley-type Recognizer for Dependency Grammar.

In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*. pp. 723–728.
- Maruyama, Hiroshi (1990). Structural Disambiguation with Constraint Propagation.
  In *Proceedings of the 28th Meeting of the Association for Computational Linguistics (ACL)*. pp. 31–38.
- McDonald, Ryan, Koby Crammer and Fernando Pereira (2005a). Online Large-Margin Training of Dependency Parsers.
  In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*. pp. 91–98.
- McDonald, Ryan, Kevin Lerman and Fernando Pereira (2006). Multilingual Dependency Analysis with a Two-Stage Discriminative Parser.
  In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL)*.
- McDonald, Ryan and Fernando Pereira (2006). Online Learning of Approximate Dependency Parsing Algorithms.
  In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. pp. 81–88.
- McDonald, Ryan, Fernando Pereira, Kiril Ribarov and Jan Hajič (2005b). Non-Projective Dependency Parsing using Spanning Tree Algorithms.

In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*. pp. 523–530.
- Mel'čuk, Igor (1988). *Dependency Syntax: Theory and Practice*.
  State University of New York Press.
- Menzel, Wolfgang and Ingo Schröder (1998). Decision Procedures for Dependency Parsing Using Graded Constraints.
  In Sylvain Kahane and Alain Polguère (eds.), *Proceedings of the Workshop on Processing of Dependency-Based Grammars*. pp. 78–87.
- Neuhaus, Peter and Norbert Bröker (1997). The Complexity of Recognition of Linguistically Adequate Dependency Grammars.
  In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL) and the 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. pp. 337–343.
- Nilsson, Jens, Joakim Nivre and Johan Hall (2006). Graph Transformations in Data-Driven Dependency Parsing.
  In *Proceedings of COLING-ACL*.
- Nivre, Joakim (2003). An Efficient Algorithm for Projective Dependency Parsing.
  In Gertjan Van Noord (ed.), *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*. pp. 149–160.

- Nivre, Joakim (2006). Constraints on Non-Projective Dependency Graphs.
  In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. pp. 73–80.
- Nivre, Joakim, Johan Hall and Jens Nilsson (2004). Memory-Based Dependency Parsing.
  In Hwee Tou Ng and Ellen Riloff (eds.), *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL)*. pp. 49–56.
- Nivre, Joakim, Johan Hall, Jens Nilsson, Gülsen Eryiğit and Svetoslav Marinov (2006). Labeled Pseudo-Projective Dependency Parsing with Support Vector Machines.
  In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL)*.
- Nivre, Joakim and Jens Nilsson (2005). Pseudo-Projective Dependency Parsing.
  In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*. pp. 99–106.
- Nivre, Joakim and Mario Scholz (2004). Deterministic Dependency Parsing of English Text.
  In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*. pp. 64–70.
- Schröder, Ingo (2002). Natural Language Parsing with Graded Constraints.

References

  Ph.D. thesis, Hamburg University.
- Sgall, Petr, Eva Hajičová and Jarmila Panevová (1986). *The Meaning of the Sentence in Its Pragmatic Aspects*.
  Reidel.
- Sleator, Daniel and Davy Temperley (1991). *Parsing English with a Link Grammar*.
  Tech. Rep. CMU-CS-91-196, Carnegie Mellon University, Computer Science.
- Tapanainen, Pasi and Timo Järvinen (1997). A non-projective dependency parser.
  In *Proceedings of the 5th Conference on Applied Natural Language Processing*. pp. 64–71.
- Tesnière, Lucien (1959). *Éléments de syntaxe structurale*.
  Editions Klincksieck.
- Yamada, Hiroyasu and Yuji Matsumoto (2003). Statistical Dependency Analysis with Support Vector Machines.
  In Gertjan Van Noord (ed.), *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*. pp. 195–206.
- Zwicky, A. M. (1985). Heads.
  Journal of Linguistics 21, 1–29.

Dependency Parsing 70(70)