# Introduction to Data-Driven Dependency Parsing

Introductory Course, ESSLLI 2007

Ryan McDonald[1] Joakim Nivre[2]

[1]Google Inc., New York, USA
E-mail: ryanmcd@google.com

[2]Uppsala University and Växjö University, Sweden
E-mail: nivre@msi.vxu.se

# Overview of the Course

- Dependency parsing (Joakim)
- Machine learning methods (Ryan)
- Transition-based models (Joakim)
- Graph-based models (Ryan)
- **Loose ends** (Joakim, Ryan):
  - Other approaches
  - Empirical results
  - Available software

# Other Approaches – Overview

- Graph-based methods – new developments
- Transition-based methods – new developments
- Ensemble methods
- Constraint-based methods
- Phrase structure parsing
- Unsupervised parsing

# Graph-based Methods

- Last lecture we discussed arc-factored models
- Models are inherently local
  - Local feature scope
  - Local structural constraints
- This is a strong assumption!!
- Question: how do we incorporate non-local information?

# Integer Linear Programming

- Often intractable inference problems can be written as Integer Linear Programming (ILP) problems
- ILP's are optimization problems with linear objectives and constraints
- Non-projective parsing with global constraints can be written as an ILP [Riedel and Clarke 2006]
- ILP's are still NP-hard, but have well known branch-and-bound solutions
- First, let's define a set of **binary variables**
  - $a_{ij}^k \in \{0,1\}$ is 1 if arc $(i,j,k)$ is in the dependency graph
  - $\boldsymbol{a}$ is the vector of all variables $a_{ij}^k$

# Integer Linear Programming

- We can define the arc-factored parsing problem as the following objective function

$$\arg\max_{\boldsymbol{a}} \sum_{i,j,k} \log w_{ij}^k \cdot a_{ij}^k$$

such that:

$$\forall j > 0,\ \textstyle\sum_{i,k} a_{ij}^k = 1 \qquad \text{(single head)}$$

$$\textstyle\sum_{i,k} a_{i0}^k = 0 \qquad (w_0 \text{ is root})$$

$$\forall \text{ cycles } C,\ \textstyle\sum_{(i,j,k)\in C} a_{ij}^k \leq |C| - 1 \qquad \text{(no cycles)}$$

- This is an ILP!!
- Linear objective
- Linear constraints over integer variables

# Integer Linear Programming

- [Riedel and Clarke 2006] showed that this formulation allows for non-local constraints
- e.g., a verb can only have a single subject

$$\forall w_i \text{ that are verbs}$$

$$\sum_j a_{ij}^{sbj} \leq 1$$

- This is non-local since we are forcing constraints on all the modifiers of $w_i$
- [Riedel and Clarke 2006] also includes constraints on co-ordination as well as projectivity if desired
- Is this still data-driven?

# Sampling Methods

- Used for dependency parsing with global features by [Nakagawa 2007]
- Define a conditional log-linear probability model

$$P(G|x) = \frac{1}{Z_x} e^{\mathbf{w} \cdot \mathbf{f}(G)}$$

- $Z_x = \sum_{G'} e^{\mathbf{w} \cdot \mathbf{f}(G')}$
- $\mathbf{f}(G)$ is a global feature map – can contain global features of dependency graph
- i.e., does not necessarily factor by the arcs

# Sampling Methods

- $\arg\max_G P(G|x)$ cannot be solved efficiently
- Assume we have $N$ samples from the distribution $P(G|x)$
  - Can be found efficiently with Gibbs sampling
  - Call them $G_1, \ldots, G_N$
- We want **marginal distribution** of the arc $(i, j, k)$, $\mu_{ij}^k$

$$\mu_{ij}^k \approx \sum_{t=1}^{N} P(G_t|x) \mathbb{1}[(i,j,k) \in G_t] \approx \frac{1}{N} \sum_{t=1}^{N} \mathbb{1}[(i,j,k) \in G_t]$$

- Since $N$ should be a managable size, this can be calculated
- Set arc weights $w_{ij}^k = \mu_{ij}^k$ and **find the MST**
- $\mathbf{w}$ is found using Monte Carlo sampling

# Transition-based Methods

- Transition-based models may suffer from
  - Error propagation because of greedy inference,
  - Label bias because of local training.
- Recent developments seek to remedy this:
  - Beam search instead of greedy best-first search
    [Johansson and Nugues 2006, Duan et al. 2007]
  - Globally trained probabilistic model
    [Johansson and Nugues 2007, Titov and Henderson 2007]

# Generative Model [Titov and Henderson 2007]

- Probabilistic model defined in terms of transitions:

$$P(G_{c_m}) = P(c_0, c_1, \ldots, c_m) = \prod_{i=1}^{m} P(c_i | c_0, \ldots, c_{i-1})$$

- Similar to HMMs
- Transition system of [Nivre 2003] with two modifications:
  - Splits Right-Arc into Right-Arc′ and Shift
  - Adds transitions for generating words (generative model)
- $P(c_i | c_0, c_1, \ldots, c_{i-1})$ modeled by neural network approximating Incremental Sigmoid Belief Network (ISBN)
  - Belief network hidden layer acts as feature selection algorithm
- Parsing with heuristic beam search

# Ensemble Methods

- Input: Sentence $x = w_0, w_1, \ldots, w_n$
- Input: Output of $N$ different parsers for $x$, $G_1, G_2, \ldots, G_N$
- Output: A single dependency graph $G$ for sentence $x$

**Question**

How do we combine the the parsers and their outputs to create a new and better parse for sentence $x$?

- Assumption: The $N$ parsers make different mistakes
- Assumption: All of the $N$ parsers are relatively **strong**

# Ensemble Methods [Sagae and Lavie 2006]

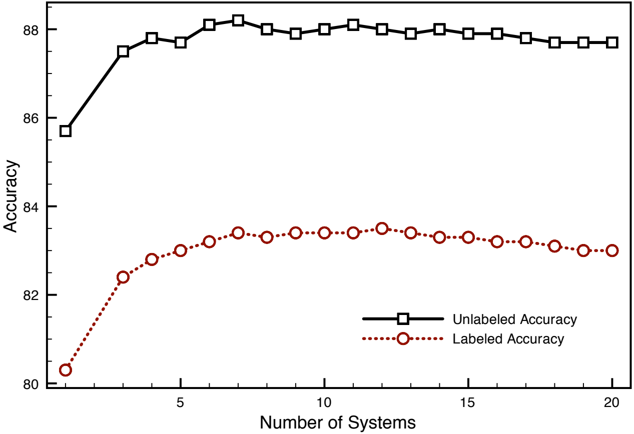
- Simple but elegant solution
- Use arc-factored graph-based models
- Set arc weights equal to the number of parsers that predicted that arc

$$w_{ij}^{k} = e^{\sum_i \alpha_i \times \mathbb{1}[(i,j,k) \in G_i]}$$

- $\alpha_i$ usually equals 1, but can be modified if prior knowledge exists
- **Solution**: Find MST for $G_x$ with the above weights
- The resulting graph has on average the arcs that were preferred by most systems

# Ensemble Methods [Sagae and Lavie 2006]

- Example: ensemble of parsers from this years CoNLL shared-task



# Constraint-based Methods

- Statistical constraint dependency parsing in two steps [Wang and Harper 2004]:
  1. Supertagging using a trigram Hidden Markov Model to assign the top $n$-best constraints to an input sentence $x$.
  2. Stack-based, best-first search to build the most probable dependency graph given the constraints.
- Anytime transformation-based parsing with constraints [Foth and Menzel 2006]:
  1. Use data-driven transition-based parser to derive initial dependency graph.
  2. Use graph transformations to improve score relative to weighted constraints.

# Phrase Structure Parsing

- Phrase structure parsers used for dependency parsing:
  1. Transform training data from dependencies to phrase structure
  2. Train a parser on the transformed structures
  3. Parse new sentences with the trained parser
  4. Transform parser output from phrase structure to dependencies
- Example:
  - Parsing Czech with the Collins and Charniak parsers
    [Collins et al. 1999, Hall and Novák 2005]
- Note:
  - Both of these parsers internally extract dependencies from phrase structures.

# Unsupervised Parsing

- Often we do not have a large corpus with annotated dependency graphs
- Can we still learn to parse dependencies from unlabeled data?
- There has been much research along these lines lately
  - Lexical attraction [Yuret 1998]
  - Grammatical bi-grams [Paskin 2001]
  - Top-down generative models [Klein and Manning 2004]
  - Contrastive estimation [Smith and Eisner 2005]
  - Non-projective examples [McDonald and Satta 2007]

# Empirical Results – Overview

- Evaluation metrics
- Benchmarks:
  - Penn Treebank (Wall Street Journal)
  - Prague Dependency Treebank
- CoNLL 2006 shared task [Buchholz and Marsi 2006]:
  - 19 parsers for 13 languages
  - Error analysis for the two top systems [McDonald and Nivre 2007]
- CoNLL 2007 shared task [Nivre et al. 2007]:
  - 23 parsers for 10 languages
  - Domain adaptation for English

# Evaluation Metrics

- Per token:
  - Labeled attachment score (LAS):
    - Percentage of tokens with correct head and label
  - Unlabeled attachment score (UAS):
    - Percentage of tokens with correct head
  - Label accuracy (LA):
    - Percentage of tokens with correct label
- Per sentence:
  - Labeled complete match (LCM):
    - Percentage of sentences with correct labeled graph
  - Unlabeled complete match (UCM):
    - Percentage of sentences with correct unlabeled graph

# State of the Art – English

- Penn Treebank (WSJ) converted to dependency graphs
  - Transition-based parsers
    [Yamada and Matsumoto 2003, Isozaki et al. 2004]
  - Graph-based parsers
    [McDonald et al. 2005a, McDonald and Pereira 2006]
  - Ensemble parser [Sagae and Lavie 2006, McDonald 2006]
  - Phrase structure parsers [Collins 1999, Charniak 2000]

| Parser | UAS | UCM |
|---|---|---|
| McDonald | 93.2 | 47.1 |
| Sagae and Lavie | 92.7 | – |
| 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 |

# State of the Art – Czech

- Prague Dependency Treebank (PDT)
  - Pseudo-projective transition-based parser [Nilsson et al. 2006]
  - Non-projective spanning tree parser [McDonald et al. 2005b]
  - Approximate second-order spanning tree parser [McDonald and Pereira 2006]
  - Phrase structure projective (Charniak, Collins)
  - Phrase structure (Charniak) + corrective modeling [Hall and Novák 2005]

| **Parser** | **UAS** | **UCM** |
|---|---|---|
| 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 | – |
| Collins | 81.8 | – |

# CoNLL Shared Task 2006

- Multilingual dependency parsing:
  - Train a single parser on data from thirteen languages
  - Gold standard annotation (postags, lemmas, etc.)
  - Main evaluation metric: LAS
- Results:
  - 19 systems, 17 described in [Buchholz and Marsi 2006]
  - Considerable variation across languages (top scores):
    - Japanese: 91.7%
    - Turkish: 65.7%
  - Best systems:
    - MSTParser (graph-based) [McDonald et al. 2006]
    - MaltParser (transition-based) [Nivre et al. 2006]

# MSTParser and MaltParser

| | 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 |
| Overall | 80.83 | 80.75 |

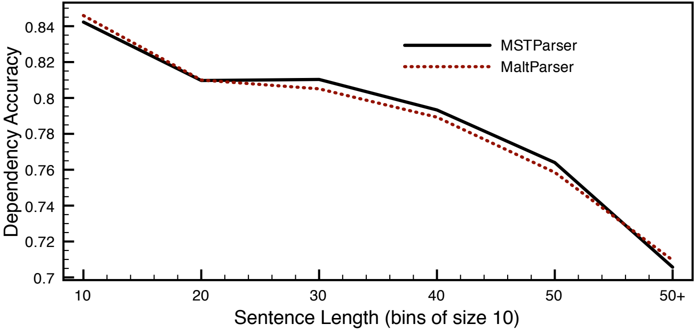
# Comparing the Models

- **Inference**:
  - Exhaustive (MSTParser)
  - Greedy (MaltParser)
- **Training**:
  - Global structure learning (MSTParser)
  - Local decision learning (MaltParser)
- **Features**:
  - Local features (MSTParser)
  - Rich decision history (MaltParser)
- **Fundamental trade-off:**
  - Global learning and inference vs. rich feature space

# Error Analysis

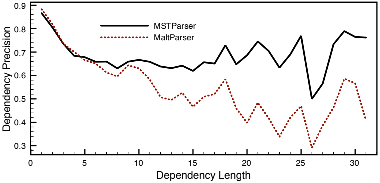
- Aim:
  - Relate parsing errors to linguistic and structural properties of the input and predicted/gold standard dependency graphs
- Three types of factors:
  - Length factors: sentence length, dependency length
  - Graph factors: tree depth, branching factor, non-projectivity
  - Linguistic factors: part of speech, dependency type
- Statistics:
  - Labeled accuracy, precision and recall
  - Computed over the test sets for all 13 languages

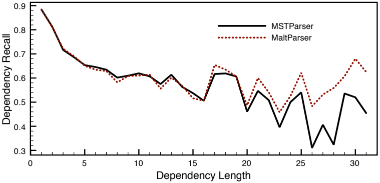
# Sentence Length




- MaltParser is more accurate than MSTParser for short sentences (1–10 words) but its performance degrades more with increasing sentence length.

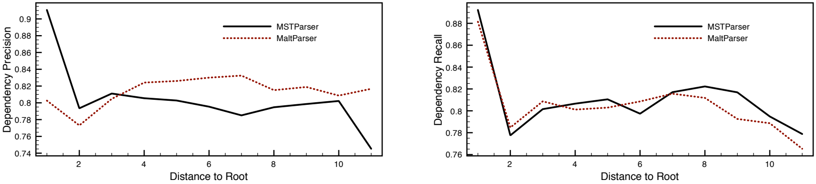
# Dependency Length



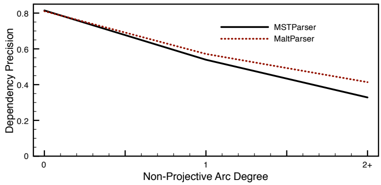

- MaltParser is more precise than MSTParser for short dependencies (1–3 words) but its performance degrades drastically with increasing dependency length ($>$ 10 words).
- MSTParser has more or less constant precision for dependencies longer than 3 words.
- Recall is very similar across systems.

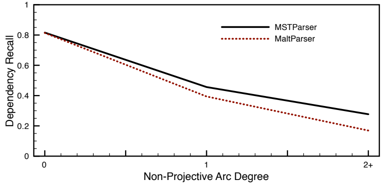
# Tree Depth (Distance to Root)




- MSTParser is much more precise than MaltParser for dependents of the root and has roughly constant precision for depth $> 1$, while MaltParser's precision improves with increasing depth (up to 7 arcs).
- Recall is very similar across systems.

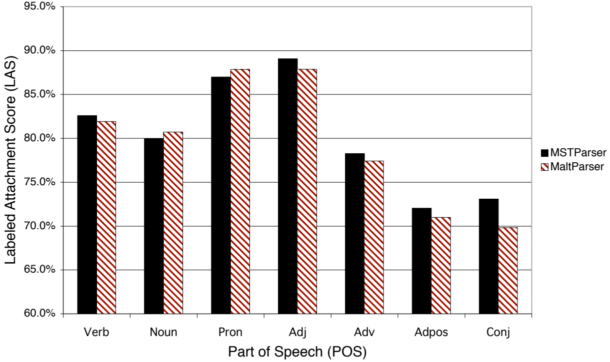
# Degrees of Non-Projectivity



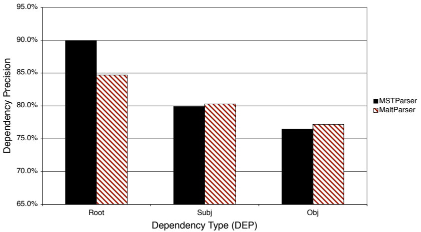

- Degree of a dependency arc $(i, j, k)$ = The number of words in the span $\min(i, j), \ldots, \max(i, j)$ that are not descendants of $i$ and have their head outside the span.
- MaltParser has slightly higher precision, and MSTParser slightly higher recall, for non-projective arcs (degree $> 0$).
- No system predicts arcs with a higher degree than 2.

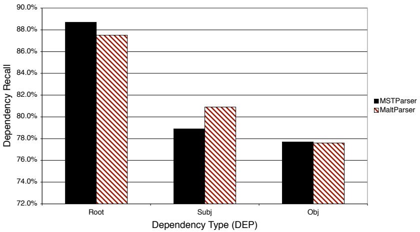
# Part of Speech




- MSTParser is more accurate for verbs, adjectives, adverbs, adpositions, and conjunctions.
- MaltParser is more accurate for nouns and pronouns.

# Dependency Type: Root, Subject, Object




- MSTParser has higher precision (and recall) for roots.
- MSTParser has higher recall (and precision) for subjects.

# Discussion

- Many of the results are indicative of the fundamental trade-off: global learning/inference versus rich features.
- Global inference improves decisions for long sentences and those near the top of graphs.
- Rich features improve decisions for short sentences and those near the leaves of the graphs.
- Important question:
  - How do we use this to improve parser performance?
- Oracle Experiments:
  - Graph-based selection: 81% → 85%
  - Arc-based selection [Sagae and Lavie 2006]: 81% → 87%

# CoNLL Shared Task 2007

- Two tracks:
  - Multilingual dependency parsing (10 languages)
  - Domain adaptation (English)
- Results (multilingual track):
  - 28 systems, 23 described in [Nivre et al. 2007]
  - A little less variation across languages (top scores):
    - English: 89.6%
    - Greek: 76.3%
  - Best systems:
    - Ensemble systems [Hall et al. 2007, Sagae and Tsujii 2007]
    - Graph-based systems with global features [Nakagawa 2007, Carreras 2007]
    - Transition-based systems with global training [Titov and Henderson 2007]

# Available Software – Overview

- Dependency Parsing Wiki:
  - `http:depparse.uvt.nl/depparse-wiki/`
- Parsers:
  - Trainable data-driven parsers
  - Parsers for specific languages (grammar-based)
- Other tools:
  - Pseudo-projective parsing
  - Evaluation software
  - Constituency-to-dependency conversion
- Data sets:
  - Dependency treebanks
  - Other treebanks with dependency conversions

# 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**
  - Graph-based spanning tree parsers with online learning
  - URL: `http://sourceforge.net/projects/mstparser`
  - Written in Java

# Trainable Parsers (2)

- Joakim Nivre's **MaltParser**
  - Transition-based parsers with MBL and SVM
  - URL: `http://w3.msi.vxu.se/~nivre/research/MaltParser.html`
  - Executable versions are available for Solaris, Linux, Windows, and MacOS (open source version in Java planned for fall 2007)
- Ivan Titov's **ISBN Dependency Parser**
  - Incremental Sigmoid Belief Network Dependency Parser
  - Transition-based inference
  - URL: `http://cui.unige.ch/~titov/idp/`
  - Written in C

# 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/`

# Other Tools

- Pseudo-projective parsing:
  - Software based on [Nivre and Nilsson 2005]
  - `http://w3.msi.vxu.se/~nivre/research/proj/0.2/doc/Proj.html`
- Evaluation software:
  - CoNLL shared tasks:
    - `http://nextens.uvt.nl/~conll/software.html`
    - `http://deppare.uvt.nl/depparse-wiki/SoftwarePage`
- Treebank conversion software:
  - CoNLL 2006 shared task treebanks:
    - `http://depparse.uvt.nl/depparse-wiki/SoftwarePage`
  - Penn Treebank:
    - `http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html`
    - `http://nlp.cs.lth.se/pennconverter/`

# Dependency Treebanks

- Arabic: Prague Arabic Dependency Treebank
- Basque: Eus3LB
- Czech: Prague Dependency Treebank
- Danish: Danish Dependency Treebank
- Greek: Greek Dependency Treebank
- Portuguese: Bosque: Floresta sintá(c)tica
- Slovene: Slovene Dependency Treebank
- Turkish: METU-Sabanci Turkish Treebank

# Other Treebanks

- Bulgarian: BulTreebank
- Catalan: CESS-ECE
- Chinese: Penn Chinese Treebank, Sinica Treebank
- Dutch: Alpino Treebank for Dutch
- English: Penn Treebank
- German: TIGER/NEGRA, TüBa-D/Z
- Hungarian: Szeged Treebank
- Italian: Italian Syntactic-Semantic Treebank
- Japanese: TüBa-J/S
- Spanish: Cast3LB
- Swedish: Talbanken05

# Summary

- State of the art in data-driven dependency parsing:
  - Transition-based models
  - Graph-based models
  - New developments (often) targeting the weaknesses of standard models
- Empirical results:
  - CoNLL shared tasks: Dependency parsing results for some twenty languages
  - Many (different) systems achieve similar accuracy, but performance varies across languages
- Available resources: Try them out!

# Dependency Treebanks (1)

- 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/`
- Eus3LB
  - ca. 50 000 words
  - Restricted availability
  - URL: `http://ixa.si.ehu.es/lxa/lkerlerroak`

# Dependency Treebanks (2)

- 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/`
- Danish Dependency Treebank
  - ca. 5 500 trees
  - Annotation based on Discontinuous Grammar [Kromann 2003]
  - Freely downloadable
  - URL: `http://www.id.cbs.dk/~mtk/treebank/`

# Dependency Treebanks (3)

- Greek Dependency Treebank
  - ca. 70 000 words
  - Restricted availability.
  - Contact ILSP, Athens, Greece.
- 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`

# Other Treebanks (1)

- BulTreebank
  - ca. 14 000 sentences
  - URL: `http://www.bultreebank.org/`
  - Dependency version available from Kiril Simov (`kivs@bultreebank.org`)
- CESS-ECE
  - ca. 500 000 words
  - Freely available for research
  - URL: `http://www.lsi.upc.edu/~mbertran/cess-ece2/`
  - Dependency version available from Toni Marti

# Other Treebanks (2)

- 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

# Other Treebanks (3)

- 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`
- Penn Treebank
  - ca. 1 million words
  - Available from LDC, license fee
  - URL: `http://www.cis.upenn.edu/~treebank/home.html`
  - Conversion to labeled dependencies: Penn2Malt, pennconverter (see above)

# Other Treebanks (4)

- 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
- 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

# Other Treebanks (5)

- Szeged Treebank
  - ca. 82 000 sentences (1.2 million words)
  - Freely available, license agreement
  - URL: `http://www.inf.u-szeged.hu/hlt`
  - Subset in dependency format (6 000 sentences)
- Italian Syntactic-Semantic Treebank
  - ca. 300 000 words
  - Available through ELDA
  - URL: `http://www.ilc.cnr.it/viewpage.php/sez=ricerca/id=874/vers=ita`
  - Dependency version available

# Other 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 (Swedish)
  - ca. 300 000 words
  - Freely downloadable
  - URL: `http://w3.msi.vxu.se/~nivre/research/Talbanken05.html`
  - Dependency version also available

## References and Further Reading


- Sabine Buchholz and Erwin Marsi. 2006.
  CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164.

- X. Carreras. 2007.
  Experiments with a high-order projective dependency parser. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

- Eugene Charniak. 2000.
  A maximum-entropy-inspired parser. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 132–139.

- Michael Collins, Jan Hajič, Lance Ramshaw, and Christoph Tillmann. 1999.
  A statistical parser for Czech. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 505–512.

- Michael Collins. 1999.
  *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

- X. Duan, J. Zhao, and B. Xu. 2007.
  Probabilistic parsing action models for multi-lingual dependency parsing. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

- Kilian A. Foth and Wolfgang Menzel. 2006.
  Hybrid parsing: Using probabilistic models as predictors for a symbolic parser. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 321–328.

- Keith Hall and Vaclav Novák. 2005.
  Corrective modeling for non-projective dependency parsing. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*, pages 42–52.

- J. Hall, J. Nilsson, J. Nivre, G. Eryi**u**git, B. Megyesi, M. Nilsson, and M. Saers. 2007.
  Single malt or blended? A study in multilingual parser optimization. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

- Hideki Isozaki, 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)*, pages 275–281.

- R. Johansson and P. Nugues. 2006.
  Investigating multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, pages 206–210.

- R. Johansson and P. Nugues. 2007.

Incremental dependency parsing using online learning. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

- D. Klein and C. Manning. 2004.
  Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proc. ACL*.
- Matthias Trautner Kromann. 2003.
  The Danish Dependency Treebank and the DTAG treebank tool. In Joakim Nivre and Erhard Hinrichs, editors, *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT)*, pages 217–220. Växjö University Press.
- Taku Kudo and Yuji Matsumoto. 2002.
  Japanese dependency analysis using cascaded chunking. In *Proceedings of the Sixth Workshop on Computational Language Learning (CoNLL)*, pages 63–69.
- Ryan McDonald and Joakim Nivre. 2007.
  Characterizing the errors of data-driven dependency parsing models. In *Proceedings of EMNLP-CoNLL 2007*.
- R. McDonald and F. Pereira. 2006.
  Online learning of approximate dependency parsing algorithms. In *Proc EACL*.
- R. McDonald and G. Satta. 2007.
  On the complexity of non-projective data-driven dependency parsing. In *Proc. IWPT*.

- R. McDonald, K. Crammer, and F. Pereira. 2005a.
  Online large-margin training of dependency parsers. In *Proc. ACL*.
- R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005b.
  Non-projective dependency parsing using spanning tree algorithms. In *Proc. HLT/EMNLP*.
- R. McDonald, K. Lerman, and F. Pereira. 2006.
  Multilingual dependency analysis with a two-stage discriminative parser. In *Proc. CoNLL*.
- R. McDonald. 2006.
  *Discriminative Training and Spanning Tree Algorithms for Dependency Parsing*. Ph.D. thesis, University of Pennsylvania.
- T. Nakagawa. 2007.
  Multilingual dependency parsing using Gibbs sampling. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.
- Jens Nilsson, Joakim Nivre, and Johan Hall. 2006.
  Graph transformations in data-driven dependency parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 257–264.
- Joakim Nivre and Jens Nilsson. 2005.

Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 99–106.

- Joakim Nivre, Johan Hall, Jens Nilsson, Gülsen Eryiugit, 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)*, pages 221–225.

- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007.
  The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*.

- Joakim Nivre. 2003.
  An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160.

- M.A. Paskin. 2001.
  Cubic-time parsing and learning algorithms for grammatical bigram models. Technical Report UCB/CSD-01-1148, Computer Science Division, University of California Berkeley.

- S. Riedel and J. Clarke. 2006.
  Incremental integer linear programming for non-projective dependency parsing. In *Proc. EMNLP*.

- K. Sagae and A. Lavie. 2006.
  Parser combination by reparsing. In *Proc. HLT/NAACL*.
- K. Sagae and J. Tsujii. 2007.
  Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.
- N. Smith and J. Eisner. 2005.
  Guiding unsupervised grammar induction using contrastive estimation. In *Working Notes of the International Joint Conference on Artificial Intelligence Workshop on Grammatical Inference Applications*.
- I. Titov and J. Henderson. 2007.
  Fast and robust multilingual dependency parsing with a generative latent variable model. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.
- Wen Wang and Mary P. Harper. 2004.
  A statistical constraint dependency grammar (CDG) parser. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together (ACL)*, pages 42–29.
- Hiroyasu Yamada and Yuji Matsumoto. 2003.
  Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 195–206.
- D. Yuret. 1998.

*Discovery of linguistic relations using lexical attraction*. Ph.D. thesis, MIT.