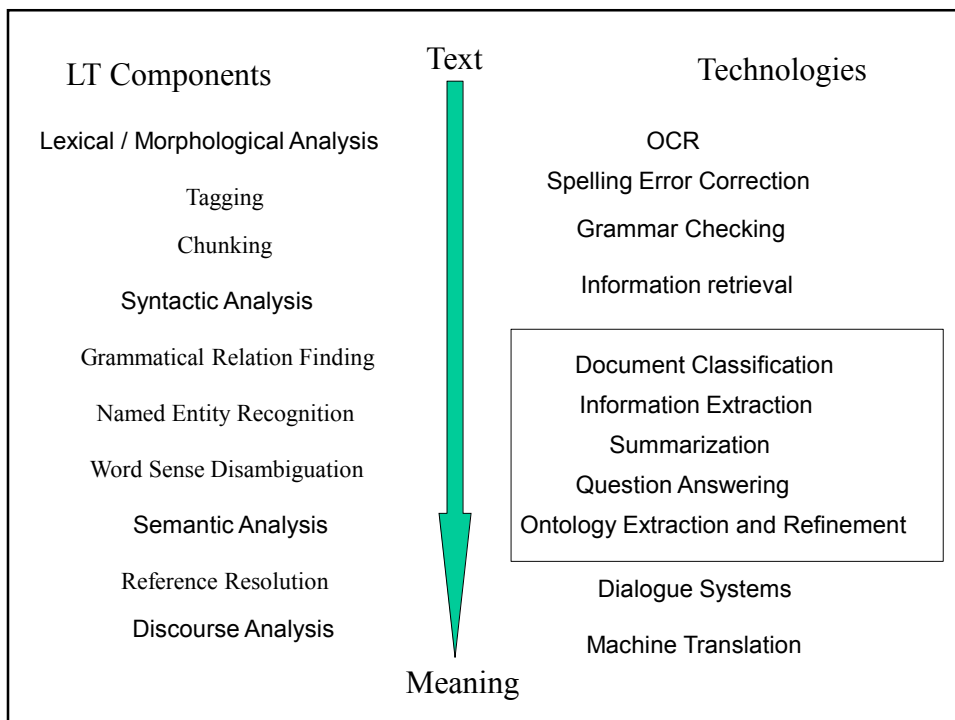


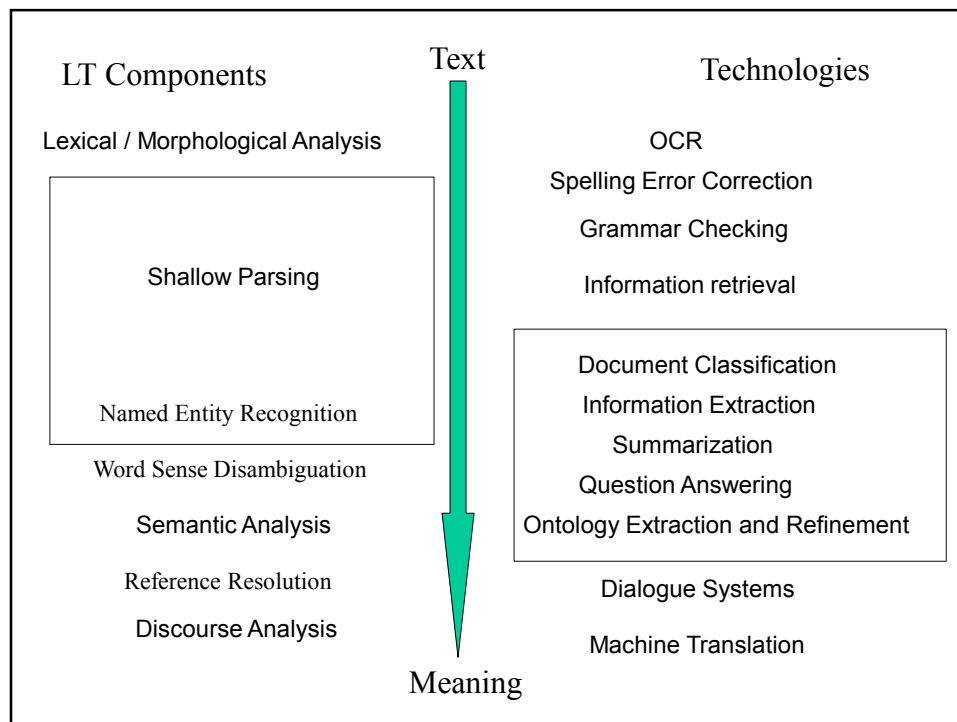
## Shallow Analysis: Light Parsing & Named Entity Extraction

Stephan Busemann

LT lab, DFKI

(based on slides by  
Günter Neumann, Steven Bird, Karin Haenelt)





## From POS tagging to Information Extraction

- **POS tagging**  
The/Det woman/NN will/MD give/VB Mary/NNP a/Det book/NN
- **NP chunking**  
The/NP1 woman/NP1 will/VP1 give/VP1 Mary/NP2 a/NP3 book/NP3
- **Relation Finding**  
[NP1-SUBJ the woman ] [VP1 will give ] [NP2-OBJ1 Mary]  
[NP3-OBJ2 a book ]]
- **Semantic Tagging = Information Extraction**  
[GIVER the woman][will give][GIVEE Mary][GIVEN a book]
- **Semantic Tagging = Question Answering**  
Who will give Mary a book?  
[GIVER ?][will give][GIVEE Mary][GIVEN a book]

## Parsing of unrestricted text

- Complexity of parsing of unrestricted text
  - Robustness
  - Large sentences
  - Large data sources
  - Input texts are not simply sequences of word forms
    - Textual structure (e.g., enumeration, spacing, etc.)
    - Combined with structural annotation (e.g., XML tags)

## Motivations for Parsing

- Why parse sentences in the first place?
- Parsing is usually an intermediate stage
  - Builds structures that are used by later stages of processing
- Full Parsing is a sufficient but not necessary intermediate stage for many NLP tasks.
- Parsing often provides more information than we need.

## Light Parsing

Goal: assign a partial structure to a sentence.

- Simpler solution space
- Local context
- Non-recursive
- Restricted (local) domain

## Output from Light Parsing

- What kind of partial structures should light parsing construct?
- Different structures useful for different tasks:
  - Partial constituent structure  
[NP I] [VP saw [NP a tall man in the park]].
  - Prosodic segments  
[I saw] [a tall man] [in the park].
  - Content word groups  
[I] [saw] [a tall man] [in the park].

## Chunk Parsing

Goal: divide a sentence into a sequence of chunks.

- Chunks are non-overlapping regions of a text  
[I] saw [a tall man] in [the park]
- Chunks are non-recursive
  - A chunk can not contain other chunks
- Chunks are non-exhaustive
  - Not all words are included in the chunks

## Chunk Parsing Examples

- Noun-phrase chunking:
  - [I] saw [a tall man] in [the park].
- Verb-phrase chunking:
  - The man who [was in the park] [saw me].
- Prosodic chunking:
  - [I saw] [a tall man] [in the park].

## Chunks and Constituency

Constituents: [[a tall man] [ in [the park]]].

Chunks: [a tall man] in [the park].

- A constituent is part of some higher unit in the hierarchical syntactic parse
- Chunks are not constituents
  - Constituents are recursive
- But, chunks are typically sub-sequences of constituents
  - Chunks do not cross major constituent boundaries

1. [<sub>NP</sub> [<sub>NP</sub> G.K. Chesterton ], [<sub>NP</sub> [<sub>NP</sub> author ] of [<sub>NP</sub> [<sub>NP</sub> The Man ] who was [<sub>NP</sub> Thursday ] ] ] ]

2. [<sub>NP</sub> G.K. Chesterton ], [<sub>NP</sub> author ] of [<sub>NP</sub> The Man ] who was [<sub>NP</sub> Thursday ]

## Chunk Parsing: Accuracy

- Chunk parsing achieves higher accuracy than full parsing
- Smaller solution space
- Less word-order flexibility within chunks than between chunks
  - Fewer long-range dependencies
  - Less contextual dependence
- Better locality
- No need to resolve ambiguity
- Less error propagation

## Chunk Parsing: Domain Specificity

- Chunk parsing is less domain specific than full parsing
- Dependencies on lexical/semantic information tend to occur at levels “higher” than chunks:
  - Attachment
  - Argument selection
  - Movement
- Fewer stylistic differences with chunks

## Psycholinguistic Motivations

Chunk parsing is psycholinguistically motivated

- Chunks are processing units
  - Humans tend to read texts one chunk at a time
  - Eye movement tracking studies
- Chunks are phonologically marked
  - Pauses
  - Stress patterns
- Chunking might be a first step in full parsing
  - Integration of shallow and deep parsing

## Chunk Parsing: Efficiency

Chunk parsing is more efficient than full parsing

- Smaller solution space
- Relevant context is small and local
- Chunks are non-recursive
- Chunk parsing can be implemented with a finite state machine
  - Fast (linear)
  - Low memory requirement (no stacks)
- Chunk parsing can be applied to a very large text sources (e.g., the web)

## Chunk Parsing Techniques

- Chunk parsers usually ignore lexical content
- Only need to look at part-of-speech tags
- Techniques for implementing chunk parsing
  - Regular expression matching
  - Chinking
  - Cascaded Finite state transducers

## Regular Expression Matching

- Define a regular expression that matches the sequences of tags in a chunk
  - A simple noun phrase chunk regexp:
    - `<DT> ? <JJ> * <NN.??>`
- Chunk all matching subsequences:
  - In:
 

```
The /DT little /JJ cat /NN sat /VBD on /IN the /DT mat /NN
```
  - Out:
 

```
[The /DT little /JJ cat /NN] sat /VBD on /IN [the /DT mat /NN]
```
- If matching subsequences overlap, the first one gets priority
- Regular expressions can be cascaded

## Chinking

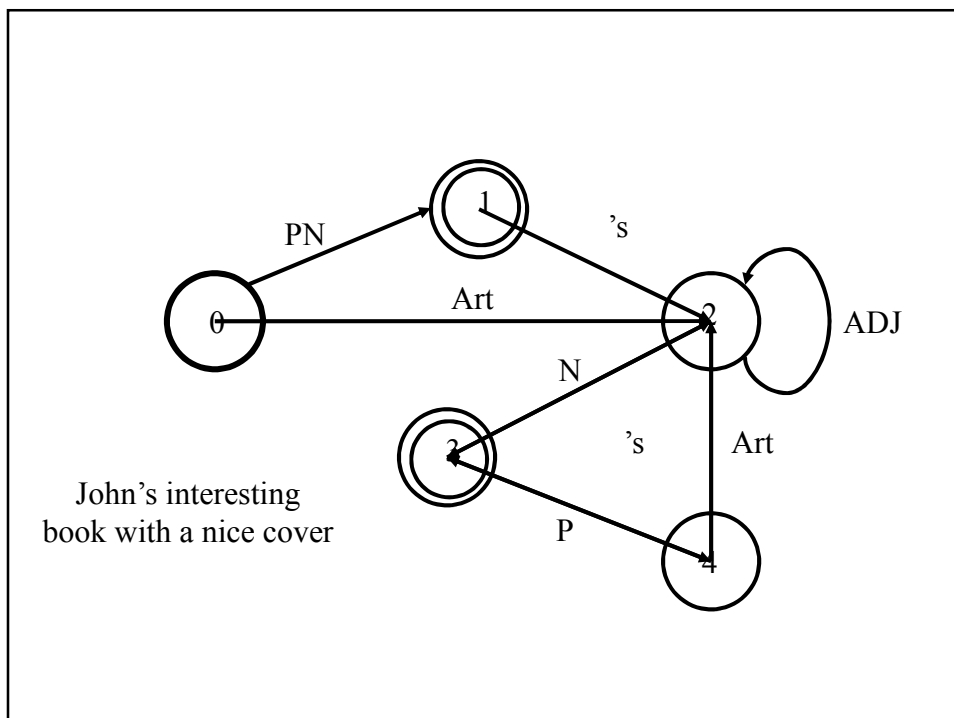
- A chink is a subsequence of the text that is not a chunk.
- Define a regular expression that matches the sequences of tags in a chink.
  - A simple chink regexp for finding NP chunks:
 

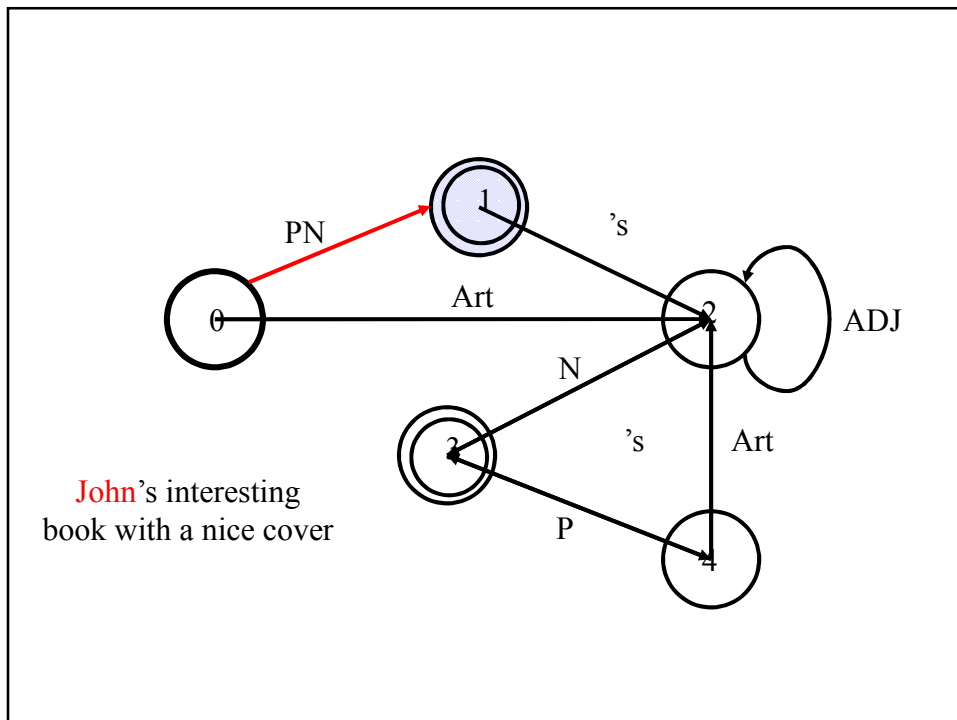
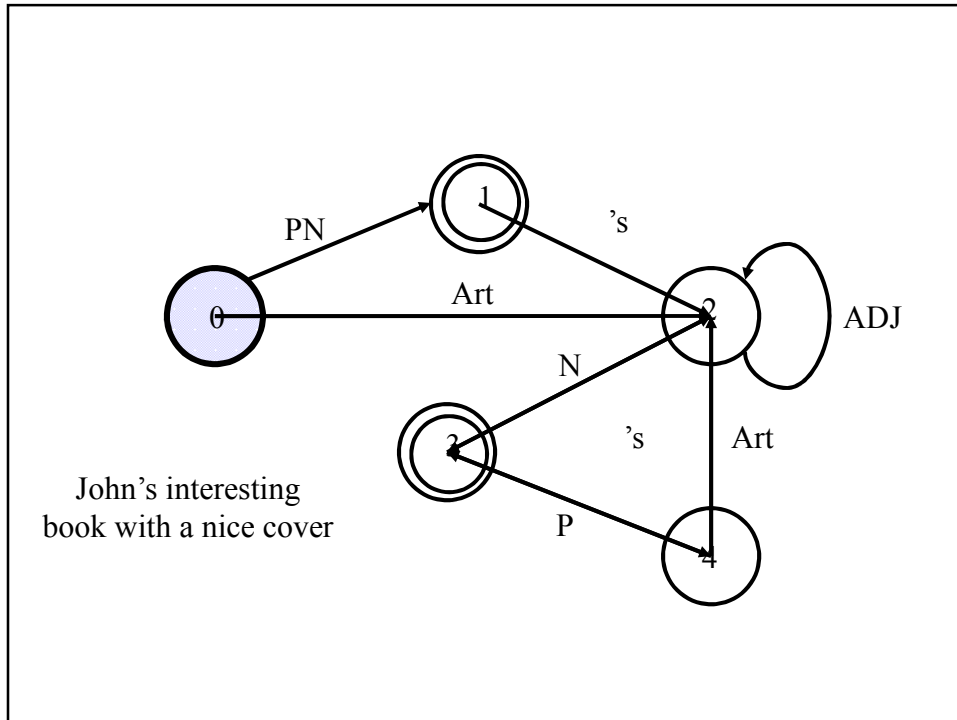
```
(<VB.?? | <IN>)+
```
- Chunk anything that is not a matching subsequence:
 

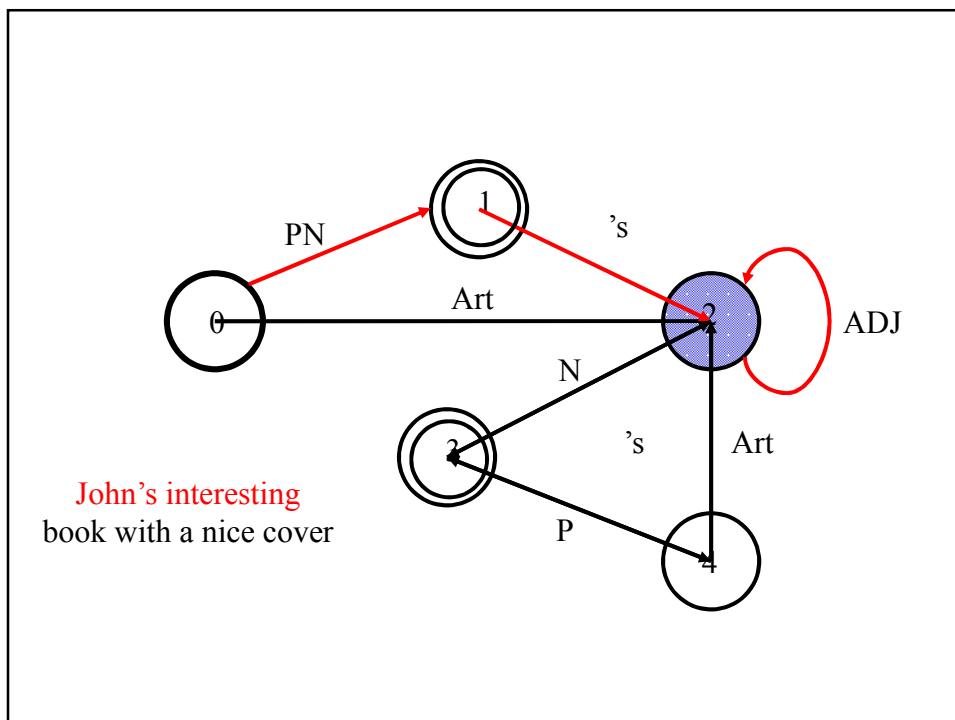
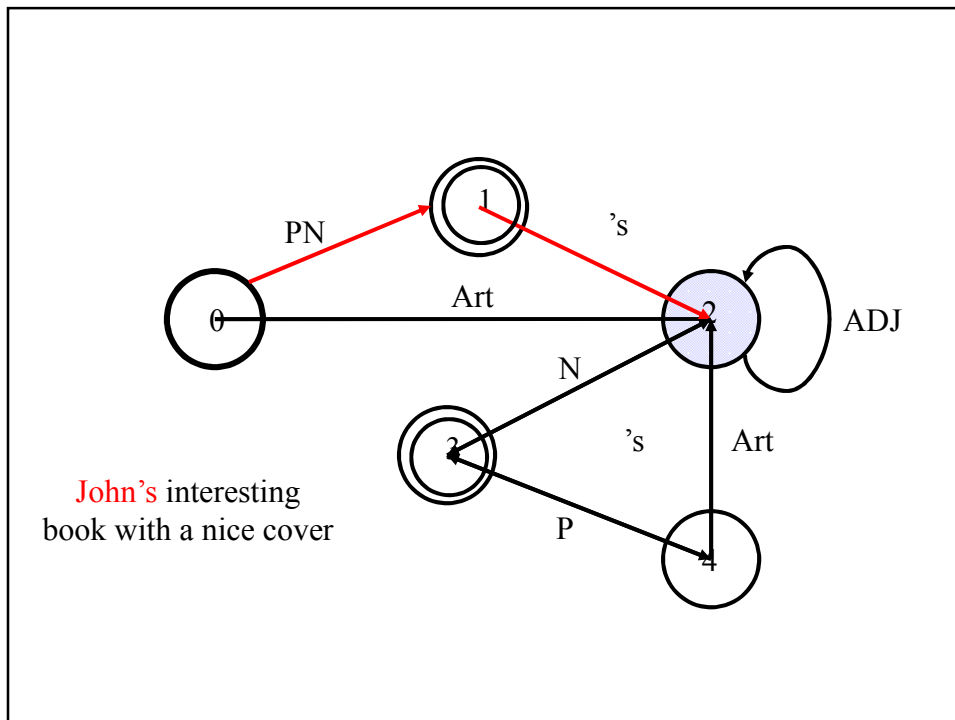
```
the/DT little/JJ cat/NN sat/VBD on /IN the /DT mat/NN
[the/DT little/JJ cat/NN] sat/VBD on /IN [the /DT mat/NN]
      chunk                chink                chunk
```

## Syntactic Structure: Partial Parsing Approaches

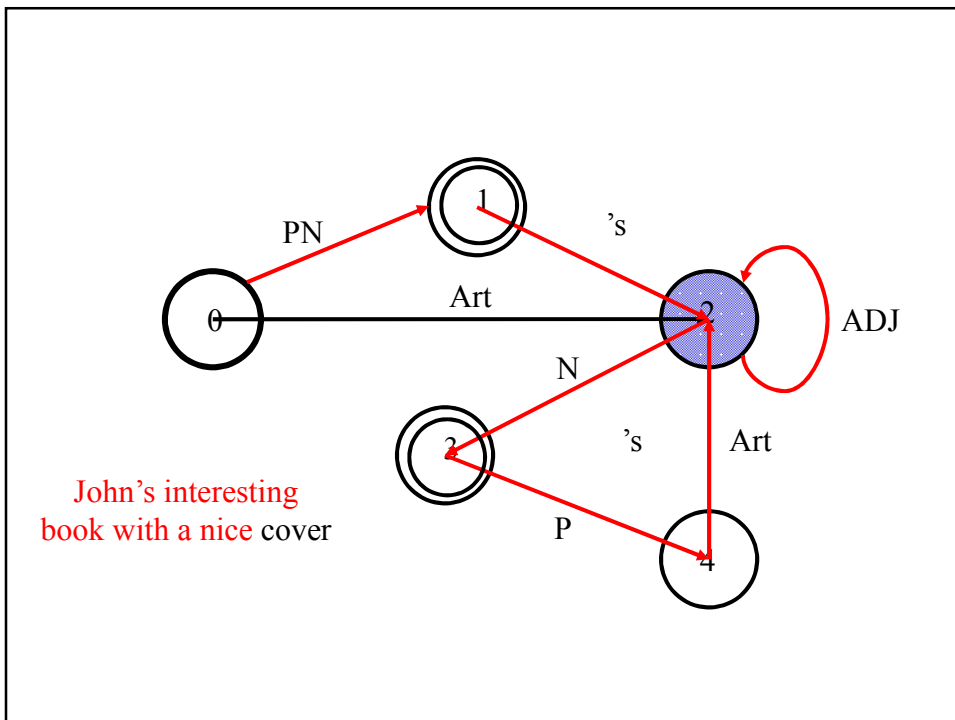
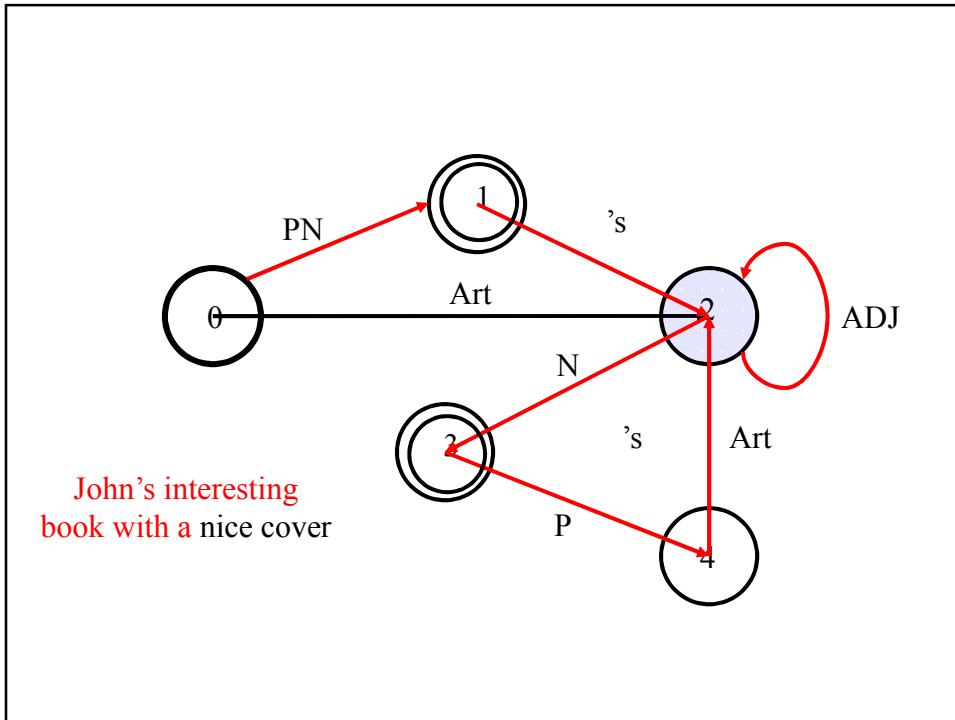
- **Finite-state approximation of sentence structures (Abney 1995)**
  - **finite-state cascades**: sequences of levels of regular expressions
  - recognition approximation: tail-recursion replaced by iteration
  - interpretation approximation: embedding replaced by fixed levels
- **Finite-state approximation of phrase structure grammars (Pereira/Wright 1997)**
  - **flattening of shift-reduce-recogniser**
  - no interpretation structure (acceptor only)
  - used in speech recognition where syntactic parsing serves to rank hypotheses for acoustic sequences
- **Finite-state approximation (Sproat 2002)**
  - **bounding of centre embedding**
  - reduction of recognition capacity
  - flattening of interpretation structure

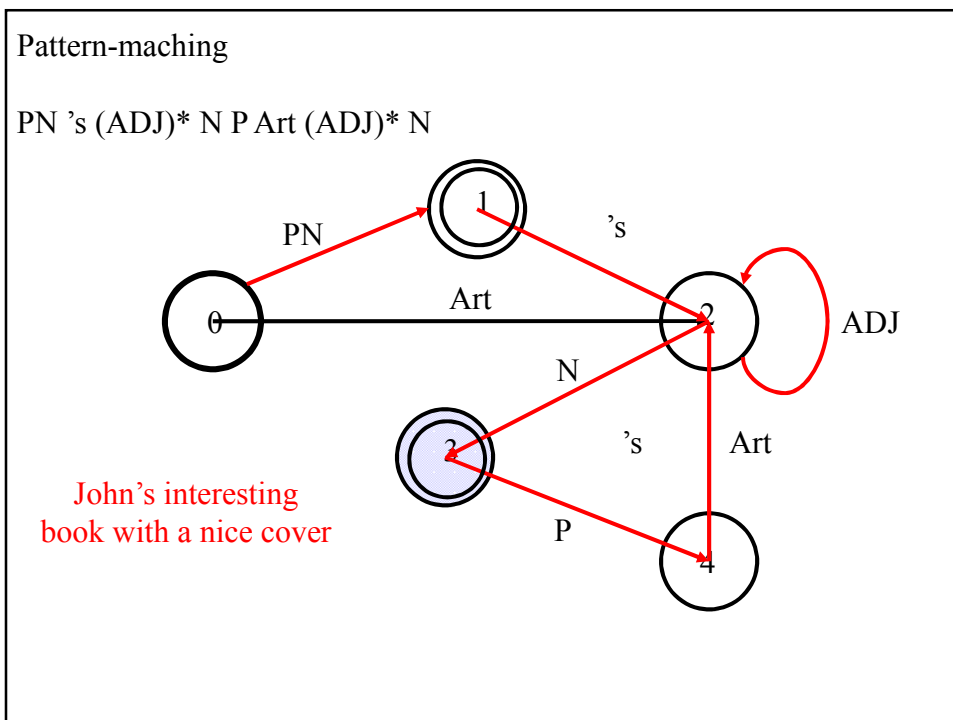
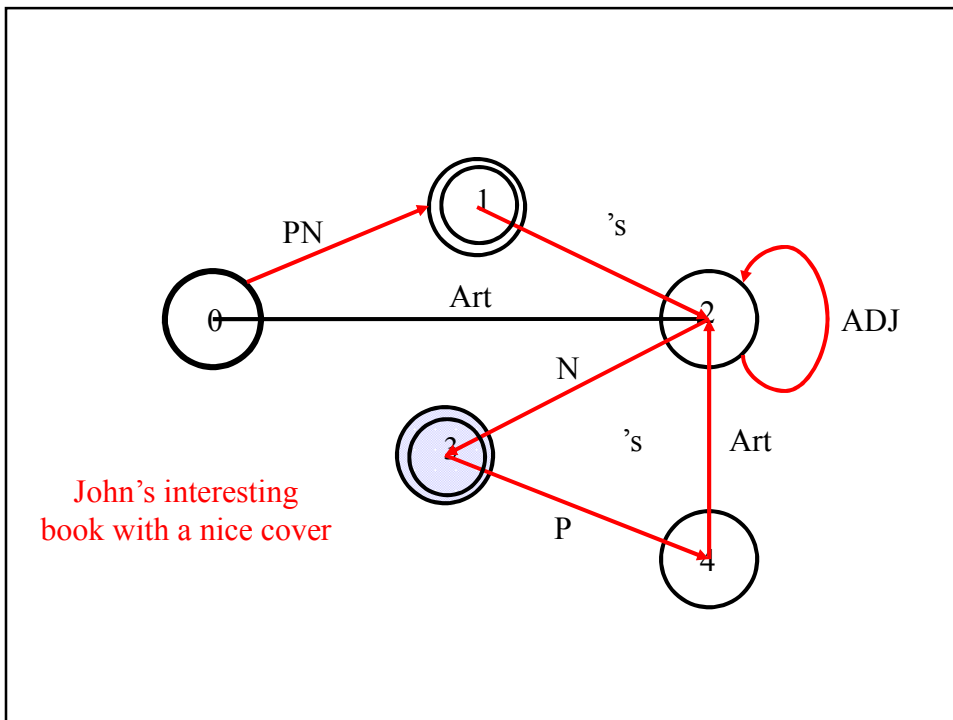






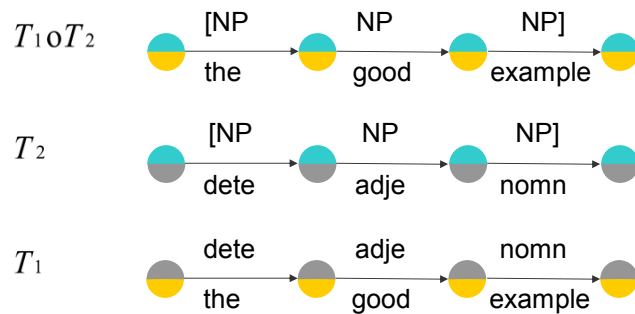






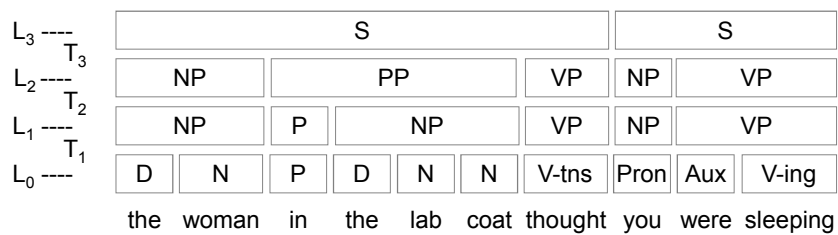
## Syntactic Structure: Finite State Cascades

- functionally equivalent to composition of transducers,
  - but without intermediate structure output
  - the individual transducers are considerably smaller than a composed transducer



## Syntactic Structure: Finite-State Cascades (Abney)

### Finite-State Cascade



### Regular-Expression Grammar

$$\begin{aligned}
 L_1: & \left\{ \begin{array}{l} NP \rightarrow D? N^* N \\ VP \rightarrow V - tns \mid Aux V - ing \end{array} \right\} \\
 L_2: & \{ PP \rightarrow P NP \} \\
 L_3: & \{ S \rightarrow NP PP^* VP PP^* \}
 \end{aligned}$$

## Syntactic Structure: Finite-State Cascades (Abney)

- A cascade consists of a sequence of levels
- Phrases at one level are built on phrases at the previous level
- No recursion:
  - phrases never contain same level or higher level phrases
- Two levels of special importance
  - chunks: non-recursive cores (NX, VX) of major phrases (NP, VP)
  - simplex clauses: embedded clauses as siblings
- Patterns:
  - reliable indicators of bits of syntactic structure

## An alternative FST cascade for German (free word order), Neumann et al.

Most partial parsing approaches following a bottom-up strategy:

### Major steps

lexical processing

including morphological analysis, POS-tagging, Named Entity recognition

phrase recognition

general nominal and prepositional phrases, verb groups

clause recognition via domain-specific templates

templates triggered by domain-specific predicates attached to relevant verbs;  
expressing domain-specific selectional restrictions for possible argument fillers

Bottom-up chunk parsing

perform clause recognition after phrase recognition is completed

However a bottom-up strategy showed to be problematic in case of German free text processing

Crucial properties of German

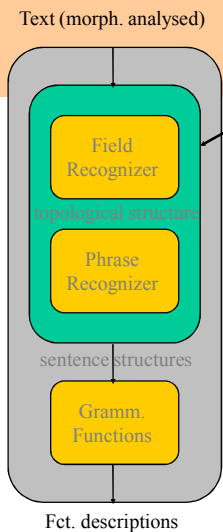
1. highly ambiguous morphology (e.g., case for nouns, tense for verbs)
2. free word/phrase order
3. splitting of verb groups into separated parts into which arbitrary phrases and clauses can be spliced in (e.g., Der Termin **findet** morgen **statt**. The date **takes place** tomorrow.)

Main problem in case of a bottom-up parsing approach:  
Even recognition of simple sentence structure depends heavily on performance of phrase recognition.

NP ist gängige Praxis.  
[NP Die vom Bundesgerichtshof und den Wettbewerbern als Verstoß gegen das Kartellverbot geäußerte zentrale TV-Vermarktung] ist gängige Praxis.

NP ist gängige Praxis.  
[NP Central television marketing criticized by the German Federal High Court and the guards against unfair competition as being an infringement of anti-cartel legislation] is common practice.

In order to overcome these problems we propose the following two phase divide-and-conquer strategy



Divide-and-conquer strategy

1. Recognize verb groups and topological structure (fields) of sentence domain-independently;

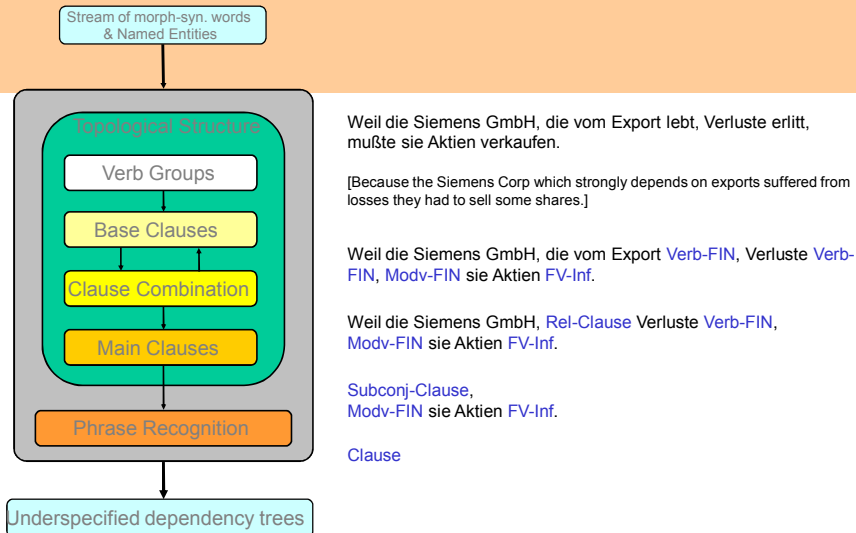
FrontField V<sub>fin</sub> MiddleField V<sub>infin</sub> RestField

2. Apply general as well as domain-dependent phrasal grammars to the identified fields of the main and sub-clauses

[CoordS [CSent Diese Angaben konnte der Bundesgrenzschutz aber nicht bestätigen], [CSent Kinkel sprach von Horrorzahlen, [Relcl denen er keinen Glauben schenke]]].

This information couldn't be verified by the Border Police, Kinkel spoke of horrible figures that he didn't believe.

## The divide-and-conquer parser is realized by means of a cascade of finite state grammars



## Semantic Analysis Sketch of Selected Approaches

- **Head-modifier-pairs**
  - mass data parsing with identifying pairs like [H: extract, M: information]
  - used in information retrieval for enriching the document index and improving retrieval efficiency (Strzalkowski/Lin/Ge/Perez-Carballo, Jose (1999)).
- **Fact extraction in fixed domains**
  - information patterns in highly standardized text types (weather forecasts, stock market reports)
  - example: biography
    - $[A-Z][a-z]^*$ ,  $[A-Z][a-z]^*$ ,  $^*[0-9]{4}^*$  in  $[A-Z][a-z]^*$ , † „ $[0-9]{4}$ “ in  $[A-Z][a-z]^*$
    - Buonarroti, Michelangelo, \*1475 in Caprese , † 1564 in Roma

## Semantic Analysis

### Sketch of Selected Approaches (cont'd)

- **Message understanding**

- filling in relational database templates from newswire texts
- approach of FASTUS <sup>1)</sup>: cascade of five transducers

- recognition of names,
- fixed form expressions,
- basic noun and verb groups
- patterns of events

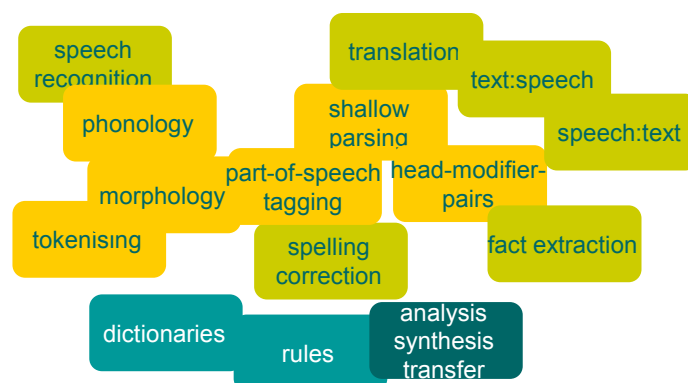
Relationship	TIE-UP
Entities	Bridgestone Sports Co. a local concern a Japanese trading house
JV Company	-
Capitalization	-

- <company> <form><joint venture> with <company>
- "Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan."
- identification of event structures that describe the same event

<sup>1)</sup> Hobbs/Appelt/Bear/Israel/Kehler/Martin/Meyers/Kameyama/Stickel/Tyson (1997)

## Final State

- Introduction
- Morphology
- Syntax
- Semantics
- Summary



## References

- Abney, Steven (1996). Tagging and Partial Parsing. In: Ken Church, Steve Young, and Gerrit Bloothoof (eds.), *Corpus-Based Methods in Language and Speech*. Kluwer Academic Publishers, Dordrecht. <http://www.vinartus.net/spa/95a.pdf>
- Abney, Steven (1996a) Cascaded Finite-State Parsing. Viewgraphs for a talk given at Xerox Research Centre, Grenoble, France. <http://www.vinartus.net/spa/96a.pdf>
- Abney, Steven (1995). Partial Parsing via Finite-State Cascades. In: *Journal of Natural Language Engineering*, 2(4): 337-344. <http://www.vinartus.net/spa/97a.pdf>
- Barton Jr., G. Edward; Berwick, Robert, C. und Eric Sven Ristad (1987). *Computational Complexity and Natural Language*. MIT Press.
- Beesley Kenneth R. und Lauri Karttunen (2003). *Finite-State Morphology*. Distributed for the Center for the Study of Language and Information. (CSLI- Studies in Computational Linguistics)
- Bod, Rens (1998). *Beyond Grammar. An Experienced-Based Theory of Language*. CSLI Lecture Notes, 88, Standford, California: Center for the Study of Information and Language
- Grefenstette, Gregory (1999). Light Parsing as Finite State Filtering. In: Kornai 1999, S. 86-94. earlier version in: *Workshop on Extended finite state models of language*, Budapest, Hungary, Aug 11-12, 1996. ECAI'96. <http://citeseer.nj.nec.com/grefenstette96light.html>
- Hobbs, Jerry; Doug Appelt, John Bear, David Israel, Andy Kehler, David Martin, Karen Meyers, Megumi Kameyama, Mark Stickel, Mabry Tyson (1997). *Breaking the Text Barrier*. FASTUS Presentation slides. SRI International. <http://www.ai.sri.com/~israel/Generic-FASTUS-talk.pdf>
- Jurafsky, Daniel und James H. Martin (2000): *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. New Jersey: Prentice Hall.
- Kornai, András (ed.) (1999). *Extended Finite State Models of Language*. (Studies in Natural Language Processing). Cambridge: Cambridge University Press.
- Koskenniemi, Kimmo (1983). Two-level morphology: a general computational model for word-form recognition and production. Publication 11, University of Helsinki. Helsinki: Department of Genral Linguistics

## References

- Kunze, Jürgen (2001). *Computerlinguistik. Voraussetzungen, Grundlagen, Werkzeuge*. Vorlesungsskript. Humboldt Universität zu Berlin. [http://www2.rz.hu-berlin.de/compling/Lehrstuhl/Skripte/Computerlinguistik\\_1/index.html](http://www2.rz.hu-berlin.de/compling/Lehrstuhl/Skripte/Computerlinguistik_1/index.html)
- Manning, Christopher D.; Schütze, Hinrich (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Mass., London: The MIT Press. <http://www.sultry.arts.usyd.edu.au/fsnlp>
- Mohri, Mehryar (1997). Finite State Transducers in Language and Speech Processing. In: *Computational Linguistics*, 23, 2, 1997, S. 269-311. <http://citeseer.nj.nec.com/mohri97finitestate.html>
- Mohri, Mehryar (1996). On some Applications of finite-state automata theory to natural language processing. In: *Journal of Natural Language Engineering*, 2, S. 1-20.
- Mohri, Mehryar und Michael Riley (2002). Weighted Finite-State Transducers in Speech Recognition (Tutorial). Teil 1: <http://www.research.att.com/~mohri/postscript/icslp.ps>, Teil 2: <http://www.research.att.com/~mohri/postscript/icslp-tut2.ps>
- G. Neumann, C. Braun and J. Piskorski (2000) *A Divide-and-Conquer Strategy for Shallow Parsing of German Free Texts* Proceedings of ANLP-2000, Seattle, Washington, pages 239-246
- Partee, Barbara; ter Meulen, Alice and Robert E. Wall (1993). *Mathematical Methods in Linguistics*. Dordrecht: Kluwer Academic Publishers.
- Pereira, Fernando C. N. and Rebecca N. Wright (1997). Finite-State Approximation of Phrase-Structure Grammars. In: Roche/Schabes 1997.
- Roche, Emmanuel und Yves Schabes (Eds.) (1997). *Finite-State Language Processing*. Cambridge (Mass.) und London: MIT Press.
- Sproat, Richard (2002). The Linguistic Significance of Finite-State Techniques. February 18, 2002. <http://www.research.att.com/~rws>
- Strzalkowski, Tomek; Lin, Fang; Ge, Jin Wang; Perez-Carballo, Jose (1999). Evaluating Natural Language Processing Techniques in Information Retrieval. In: Strzalkowski, Tomek (Ed.): *Natural Language Information Retrieval*, Kluwer Academic Publishers, Holland : 113-145
- Woods, W.A. (1970). Transition Network Grammar for Natural Language Analysis. In: *Communications of the ACM* 13: 591-602.

## Named Entity Extraction

### Machine Learning for Named Entity Extraction

## The who, where, when & how much in a sentence

- The task: identify lexical and phrasal information in text which express references to named entities NE, e.g.,
  - person names
  - company/organization names
  - locations
  - dates&times
  - percentages
  - monetary amounts
- Determination of an NE
  - Specific type according to some taxonomy
  - Canonical representation (template structure)

## Example of NE-annotated text

Delimit the named entities in a text and tag them with NE types:

```
<ENAMEX TYPE=„LOCATION“>Italy</ENAMEX>'s business world was rocked by
the announcement <TIMEX TYPE=„DATE“>last Thursday</TIMEX> that Mr.
<ENAMEX TYPE=„PERSON“>Verdi</ENAMEX> would leave his job as vice-
president of <ENAMEX TYPE=„ORGANIZATION“>Music Masters of Milan,
Inc</ENAMEX> to become operations director of
<ENAMEX TYPE=„ORGANIZATION“>Arthur Andersen</ENAMEX>.
```

- „Milan“ is part of organization name
- „Arthur Andersen“ is a company
- „Italy“ is sentence-initial – capitalization useless

## NE and Question-Answering

- Often, the expected answer type of a question is a NE
  - What was the name of the first Russian astronaut to do a spacewalk?
    - Expected answer type is PERSON
  - Name the five most important software companies!
    - Expected answer type is a list of COMPANY
  - Where is does the ESSLLI 2004 take place?
    - Expected answer type is LOCATION (subtype COUNTRY or TOWN)
  - When will be the next talk?
    - Expected answer type is DATE

## Difficulties of Automatic NEE

- NEs can't be enumerated in order to include them in dictionaries/Gazetteers
- Names are changing constantly
- Names vary in form
- Subsequent occurrences of names might be abbreviated

- list search/matching does not perform well
- context based pattern matching needed

## Difficulties for Pattern Matching Approach

Whether a phrase is a named entity, and what name class it has, depends on

- Internal structure:  
„Mr. Brandon“
- Context:  
„The new company, SafeTek, will make air bags.“
- Feiyu Xu, researcher at DFKI, Saarbrücken

## Statistical and Rule-Based NEE

- Identify a type of NE from charpos S to charpos E
  - Giuseppe Verdi, Italian composer of „Aida“  
 <NE TYPE=„PERSON“>Giuseppe Verdi</NE>
- Recognize structured entities with rule-based systems
  - Prof. Dr. Wolfgang Wahlster, CEO of DFKI GmbH  
 <NE TYPE=„PERSON“>  
   <ELTS> <NE TYPE=„TITLE“>Prof. Dr. </NE>  
     <NE TYPE=„FIRSTNAME“>Wolfgang</NE>  
     <NE TYPE=„LASTNAME“>Wahlster</NE>  
     <NE TYPE=„FUNCTION“>CEO</NE>  
     <NE TYPE=„COMPANY“>  
       <ELTS> <NE TYPE „NAME“>DFKI</NE>  
         <NE TYPE „DGNR“>GmbH</NE>  
       </ELTS></NE>  
     </ELTS></NE>  
 </ELTS> </NE>

## Different ML approaches to NEE

- Supervised learning
  - Training is based on available very large annotated corpora
  - Mainly statistics-based methods are used, e.g. HMM, MEM, connectionist models, SVM, hybrid ML-methods (cf. <http://www.cnts.ua.ac.be/conll2003/ner/> )
- Semi-supervised learning
  - Training only needs very few seeds
  - Very large un-annotated corpus, usually larger than for supervised learning
- Unsupervised Learning
  - Typical approach is clustering, e.g., cluster NEs on basis of similar context (common syntagmatic relationship). Problem: naming the clusters, e.g., using WordNet labels, cf. (Alfonseca and Mandandhar, 2004)
  - Hypernym rules, e.g., “X such as A, B, C” -> A, B, C are NEs of type X, cf. (Evans 2003)

## Semi-Supervised NEE: Sketch

- Define manually a small set of trusted seeds
- Training uses un-labeled data
- Initialize system by labeling the corpus with the analysis of the seed
- Extract and generalize patterns from the context of the seeds
- Use the patterns to further label the corpus and to extend the seed set (**bootstrapping**)
- Repeat the process until no new terms can be identified

[See lecture on Relation Extraction for more on this.]

## References for NEE

- Alfonseca, Enrique; Manandhar, S. 2002. An Unsupervised Method for General Named Entity Recognition and Automated Concept Discovery. In Proc. International Conference on General WordNet.
- D. Bikel, S. Miller, Richard Schwartz and Ralph Weischedel: "Nymble: a High-Performance Learning Name Finder" ANLP 1997.
- Chen, H. H.; Lee, J. C. 1996. Identification and Classification of Proper Nouns in Chinese Texts. In Proc. International Conference on Computational Linguistics.
- M. Collins, Y. Singer "Unsupervised Models for Named Entity Classification", EMNLP 1999.
- Drozdowski, Witold; Krieger, Hans-Ulrich; Piskorski, Jakub; Schäfer, Ulrich, Xu Feiyu. 2004. Shallow Processing with Unification and Typed Feature Structures - Foundations and Applications. *Künstliche Intelligenz 1(1)*, 17-23. ([http://www.dfki.de/lt/publication\\_show.php?id=1295](http://www.dfki.de/lt/publication_show.php?id=1295))
- Evans, Richard. 2003. A Framework for Named Entity Recognition in the Open Domain. In Proc. Recent Advances in Natural Language Processing.
- Fleischman, Michael; Hovy, E. 2002. Fine Grained Classification of Named Entities. In Proc. Conference on Computational Linguistics.
- Huang, Fei. 2005. Multilingual Named Entity Extraction and Translation from Text and Speech. Ph.D. Thesis. Pittsburgh: Carnegie Mellon University.
- Lin, Yangarber, Grishman, ICML 2003
- A. McCallum and W. Li, "Early Results for Named Entity Recognition with Conditional Random Fields, Fetures Induction and Web-Enhanced Lexicons", CoNLL 2003.
- Nadeau, David, Satoshi Sekine (2006). A survey of named entity recognition and classification, Special issue of *Linguisticæ Investigationes* 30:1 (2007) (Available from Günter Neumann, DFKI LT Lab.)
- Neumann (2007): web course page - <http://www.dfki.de/%7Eneumann/meta-ner/SoftWareProject.html>
- [http://en.wikipedia.org/wiki/Named\\_entity\\_recognition](http://en.wikipedia.org/wiki/Named_entity_recognition)
- E. Riloff, R. Jones: "Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping", AAAI 1999.
- Yangarber, Lin, Grishman, Coling 2002