Shallow Processing & Named Entity Extraction

Günter Neumann, Bogdan Sacaleanu
LT lab, DFKI

(includes modified slides from Steven Bird, Gerd Dalemans, Karin Haenelt)
LT Components

Lexical / Morphological Analysis
  - Tagging
  - Chunking

Syntactic Analysis
  - Grammatical Relation Finding
  - Named Entity Recognition
  - Word Sense Disambiguation

Semantic Analysis
  - Reference Resolution

Discourse Analysis

Applications

  OCR
  - Spelling Error Correction
  - Grammar Checking
  - Information retrieval

  Document Classification
  - Information Extraction
  - Summarization
  - Question Answering
  - Ontology Extraction and Refinement

  Dialogue Systems
  - Machine Translation
From POS tagging to IE - Classification-Based Perspective

- POS tagging
  The/ Det woman/ NN will/ MD give/ VB Mary/ NNP a/ Det book/ NN

- NP chunking
  The/ B-NP woman/ I-NP will/ B-VP give/ I-VP Mary/ B-NP a/ B-NP book/ I-NP

- Grammatical Relation Finding
  [NP-SUBJ-1 the woman ] [VP-1 will give ] [NP-I-OBJ-1 Mary ] [NP-OBJ-1 a book ]

- Semantic Tagging (as for Information Extraction)
  [Giver the woman][will give][Givee Mary][Given a book]

- Semantic Tagging (as for Question Answering)
  Who will give Mary a book?
  [Giver ?][will give][Givee Mary][Given a book]
• Complexity of parsing of unrestricted text
  – 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)
  – Various text styles, e.g., newspaper text, scientific texts, blogs, email, …
    • Demands high degree of flexibility and robustness
Motivations for Parsing

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

• Light (or “partial”) parsing
• Chunk parsing (a type of light parsing)
  – Introduction
  – Advantages
  – Implementations
• Divide-and-conquer parsing for German
Light Parsing

Goal: assign a *partial structure* to a sentence.

- Simpler solution space
- Local context
- Non-recursive
- Restricted (local) domain
What kind of partial structures should light parsing construct?

Different structures useful for different tasks:

- Partial constituent structure
  \[\text{NP I} \ [\text{VP saw} \ [\text{NP a tall man in the park}]\].

- Prosodic segments
  \[\text{I saw} \ [\text{a tall man} \ [\text{in the park}]\].

- Content word groups
  \[\text{I} \ [\text{saw} \ [\text{a tall man} \ [\text{in the park}]\].}
Goal: divide a sentence into a sequence of chunks.

- Chunks are non-overlapping regions of a text
  
  \[\text{I saw a tall man in the park}\]

- 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:**  \([ [a \text{ tall man}] \ [in \ [the \ park]] ] \).

**Chunks:**  \([a \text{ tall man}] \ \text{in} \ [the \ park]\).

- 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. \([\text{NP } \text{NP} \ G.K. \ \text{Chesterton }], \text{NP } \text{NP} \ \text{author } \text{of} \text{NP } \text{NP} \ \text{The Man } \text{who was} \text{NP } \text{NP} \ \text{Thursday } ] \)
2. \([\text{NP } \text{NP} \ G.K. \ \text{Chesterton }], \text{NP } \text{NP} \ \text{author } \text{of} \text{NP } \text{NP} \ \text{The Man } \text{who was} \text{NP } \text{NP} \ \text{Thursday }] \)
Chunk parsing achieves high accuracy

- Small solution space
- Less word-order flexibility *within* chunks than *between* chunks
  - Fewer long-range dependencies
  - Less context dependence
- Better locality
- No need to resolve ambiguity
- Less error propagation
Chunk parsing is less domain specific

- Dependencies on lexical/semantic information tend to occur at levels “higher” than chunks:
  - Attachment
  - Argument selection
  - Movement

- Fewer stylistic differences with chunks
Psycholinguistic Motivations

• 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
  – Text zooming
Chunk Parsing: Efficiency

- 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 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
• Define a regular expression that matches the sequences of tags in a chunk
  – A simple noun phrase chunk regrexp:
    • <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
• 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:
    \( (<\text{VB.}?\>) | (<\text{IN}>)+ \)

• 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
Finite State Approaches to Shallow Parsing

• 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
John’s interesting book with a nice cover
John’s interesting book with a nice cover
John’s interesting book with a nice cover
John’s interesting book with a nice cover
John’s interesting book with a nice cover
John’s interesting book with a nice cover
John’s interesting book with a nice cover
John’s interesting book with a nice cover
John’s interesting book with a nice cover
John’s interesting book with a nice cover
John’s interesting book with a nice cover
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

\[
\begin{array}{c|c|c|c}
L_0 & S & L_3 & T_3 \\
D & N & NP & PP \\
P & P & VP & NP \\
D & N & V-tns & Pron \\
N & N & Aux & V-ing \\
\end{array}
\]

Regular-Expression Grammar

\[
L_1:\left\{
\begin{array}{l}
NP \rightarrow D? N \ast N \\
VP \rightarrow V-tns | Aux V-ing \\
\end{array}
\right\}
\]

\[
L_2:\{PP \rightarrow P \ NP\}
\]

\[
L_3:\{S \ PP* \ NP \ PP* \ VP \ PP*\}
\]

NOTE:
No recursion allowed
Syntactic Structure:
Finite-State Cascades (Abney)

• 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
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 & 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 ist gängige Praxis.*

[NP Central television marketing censured by the German Federal High Court and the guards against unfair competition as 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

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

   FrontField LeftVerb MiddleField RightVerb 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

Weil die Siemens GmbH, die vom Export lebt, Verluste erlitt, mußte sie Aktien verkaufen.
*Because the Siemens Corp which strongly depends on exports suffered from losses they had to sell some shares.*

Weil die Siemens GmbH, die vom Export *Verb-FIN*, Verluste *Verb-FIN*, *Modv-FIN* sie Aktien *FV-Inf*.


*Subconj-Clause*,
*Modv-FIN* sie Aktien *FV-Inf*.
• **Chunk linking and chunk attachment (Abney)**
  – Interpretation steps in partial parsing
  – linking of hitherto unconnected structures (attachment of modifiers, prepositional phrases, determination of subject and object)
  – interpretation basis: case frames, corpus examples

• **Finite state filtering (Grefenstette, 1999)**
  – layered finite-state parser
  – groups adjacent syntactically related units
  – extracts non-adjacent n-ary grammatical relations.
  – high level specifications of regular expressions or describing the patterns to be extracted.
• **head-modifier-pairs**
  – mass data parsing with identifying pairs like [H: extraction, 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
Selected Approaches (3)

• message understanding/information extraction
  – filling in relational database templates from newswire texts
  – approach of FASTUS ¹): cascade of five transducers
    • recognition of names,
    • fixed form expressions,
    • basic noun and verb groups
    • patterns of events
      – <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

¹) Hobbs/Appelt/Bear/Israel/Kehler/Martin/Meyers/Kameyama/Stickel/Tyson (1997)
References

http://www2.rz.hu-berlin.de/compling/Lehrstuhl/Skripte/Computerlinguistik_1/index.html

http://www.sultry.arts.usyd.edu.au/fsnlp

http://citeseer.nj.nec.com/mohri97finitestate.html


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)
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 ESSLII 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

- Potential set of NE is too numerous to include in dictionaries/Gazetteers
- Names changing constantly
- Names appear in many variant forms
- Subsequent occurrences of names might be abbreviated

- list search/matching does not perform well
- context based pattern matching needed
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
NE is an interesting problem

- Productivity of name creation requires lexicon free pattern recognition
- NE ambiguity requires resolution methods
- Fine-grained NE classification needs fine-grained decision making methods
  - Taxonomy learning
- Multi-linguality
  - A text might contain NE expressions from different languages
  - New pilot challenge in ACE’2007
    - Extract all NEs mentioned in a Mandarin/Arabic text
    - Translate them to English

• Martin Marietta can be a person name or a reference to a company
• If MM is not part of an abbreviation lexicon, how do we recognize it?
  – Also by taking into account NE reference resolution.
Why Machine Learning NE?

- **System-based adaptation for new domains**
  - Fast development cycle
  - Manual specification too expensive
  - Language-independence of learning algorithms
  - NL-tools for feature extraction available, often as open-source

- **Current approaches already show near-human-like performance**
  - Can easily be integrated with externally available Gazetteers

- **High innovation potential**
  - Core learning algorithms are language independent, which supports multi-linguality
  - Novel combinations with relational learning approaches
  - Close relationship to currently developed ML-approaches of reference resolution
Different approaches of Preprocessing

• **Character-level features**
  – (Whitelaw & Patrick, CoNLL, 2003)

• **Tokenization**
  – (Bikel et al., ANLP 1997)

• **POS + lemmatization**
  – (Yangarber et al., Coling 2002)

• **Morphology**
  – (Cucerzan & Yarowsky, EMNLP 1999)

• **Full parsing**
  – (Collins & Singer, EMNLP 1999)
Different approaches

• **Supervised learning**
  – Training is based on available very large annotated corpus
  – Mainly statistical-based methods used
    • HMM, MEM, connectionists models, SVM, hybrid ML-methods (cf. http://www.cnts.ua.ac.be/conll2003/ner/)

• **Semi-supervised learning**
  – Training only needs very few seeds and
  – 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., WordNet-labels, cf. (Alfonseca and Mandandhar, 2004)
  – Hypernym rules, “X such as A, B, C” -> A,B,C are NEs of type X, cf. (Evans 2003)
Performance of supervised methods (CoNLL, 2003)*

<table>
<thead>
<tr>
<th>English</th>
<th>precision</th>
<th>recall</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>[FIJZ03]</td>
<td>88.99%</td>
<td>88.54%</td>
<td>88.76±0.7</td>
</tr>
<tr>
<td>[CN03]</td>
<td>88.12%</td>
<td>88.51%</td>
<td>88.31±0.7</td>
</tr>
<tr>
<td>[KSNM03]</td>
<td>85.93%</td>
<td>86.21%</td>
<td>86.07±0.8</td>
</tr>
<tr>
<td>[ZJ03]</td>
<td>86.13%</td>
<td>84.88%</td>
<td>85.50±0.9</td>
</tr>
<tr>
<td>[CMP03b]</td>
<td>84.05%</td>
<td>85.96%</td>
<td>85.00±0.8</td>
</tr>
<tr>
<td>[CC03]</td>
<td>84.29%</td>
<td>85.50%</td>
<td>84.89±0.9</td>
</tr>
<tr>
<td>[MMP03]</td>
<td>84.45%</td>
<td>84.90%</td>
<td>84.67±1.0</td>
</tr>
<tr>
<td>[CMP03a]</td>
<td>85.81%</td>
<td>82.84%</td>
<td>84.30±0.9</td>
</tr>
<tr>
<td>[ML03]</td>
<td>84.52%</td>
<td>83.55%</td>
<td>84.04±0.9</td>
</tr>
<tr>
<td>[BON03]</td>
<td>84.68%</td>
<td>83.18%</td>
<td>83.92±1.0</td>
</tr>
<tr>
<td>[MLP03]</td>
<td>80.87%</td>
<td>84.21%</td>
<td>82.50±1.0</td>
</tr>
<tr>
<td>[WNC03]*</td>
<td>82.02%</td>
<td>81.39%</td>
<td>81.70±0.9</td>
</tr>
<tr>
<td>[WP03]</td>
<td>81.60%</td>
<td>78.05%</td>
<td>79.78±1.0</td>
</tr>
<tr>
<td>[HV03]</td>
<td>76.33%</td>
<td>80.17%</td>
<td>78.20±1.0</td>
</tr>
<tr>
<td>[DD03]</td>
<td>75.84%</td>
<td>78.13%</td>
<td>76.97±1.2</td>
</tr>
<tr>
<td>[Ham03]</td>
<td>69.09%</td>
<td>53.26%</td>
<td>60.15±1.3</td>
</tr>
<tr>
<td>baseline</td>
<td>71.91%</td>
<td>50.90%</td>
<td>59.61±1.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>German</th>
<th>precision</th>
<th>recall</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>[FIJZ03]</td>
<td>83.87%</td>
<td>63.71%</td>
<td>72.41±1.3</td>
</tr>
<tr>
<td>[KSNM03]</td>
<td>80.38%</td>
<td>65.04%</td>
<td>71.90±1.2</td>
</tr>
<tr>
<td>[ZJ03]</td>
<td>82.00%</td>
<td>63.03%</td>
<td>71.27±1.5</td>
</tr>
<tr>
<td>[MMP03]</td>
<td>75.97%</td>
<td>64.82%</td>
<td>69.96±1.4</td>
</tr>
<tr>
<td>[CMP03b]</td>
<td>75.47%</td>
<td>63.82%</td>
<td>69.15±1.3</td>
</tr>
<tr>
<td>[CC03]</td>
<td>84.29%</td>
<td>63.82%</td>
<td>68.88±1.3</td>
</tr>
<tr>
<td>[MMP03]</td>
<td>75.61%</td>
<td>62.46%</td>
<td>68.41±1.4</td>
</tr>
<tr>
<td>[ML03]</td>
<td>75.97%</td>
<td>61.72%</td>
<td>68.11±1.4</td>
</tr>
<tr>
<td>[MLP03]</td>
<td>69.37%</td>
<td>66.21%</td>
<td>67.75±1.4</td>
</tr>
<tr>
<td>[CMP03a]</td>
<td>77.83%</td>
<td>58.02%</td>
<td>66.48±1.5</td>
</tr>
<tr>
<td>[WNC03]</td>
<td>75.20%</td>
<td>59.35%</td>
<td>66.34±1.3</td>
</tr>
<tr>
<td>[CN03]</td>
<td>76.83%</td>
<td>57.34%</td>
<td>65.67±1.4</td>
</tr>
<tr>
<td>[HV03]</td>
<td>71.15%</td>
<td>56.55%</td>
<td>63.02±1.4</td>
</tr>
<tr>
<td>[DD03]</td>
<td>63.93%</td>
<td>51.86%</td>
<td>57.27±1.6</td>
</tr>
<tr>
<td>[WP03]</td>
<td>71.05%</td>
<td>44.11%</td>
<td>54.43±1.4</td>
</tr>
<tr>
<td>[Ham03]</td>
<td>63.49%</td>
<td>38.25%</td>
<td>47.74±1.5</td>
</tr>
<tr>
<td>baseline</td>
<td>31.86%</td>
<td>28.89%</td>
<td>30.30±1.3</td>
</tr>
</tbody>
</table>

Produced by a system which only identified entities which had a unique class in the training data.

*http://www.cnts.ua.ac.be/conll2003/ner/
### Main features used by CoNLL 2003 systems

<table>
<thead>
<tr>
<th></th>
<th>lex</th>
<th>pos</th>
<th>aff</th>
<th>pre</th>
<th>ort</th>
<th>gaz</th>
<th>chu</th>
<th>pat</th>
<th>cas</th>
<th>tri</th>
<th>bag</th>
<th>quo</th>
<th>doc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Florian</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Chieu</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Klein</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Zhang</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Carreras (a)</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Curran</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mayfield</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Carreras (b)</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>McCallum</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Bender</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Munro</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Wu</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Whitelaw</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hendrickx</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>De Meulder</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hammerton</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3: Main features used by the sixteen systems that participated in the CoNLL-2003 shared task sorted by performance on the English test data. Aff: affix information (n-grams); bag: bag of words; cas: global case information; chu: chunk tags; doc: global document information; gaz: gazetteers; lex: lexical features; ort: orthographic information; pat: orthographic patterns (like Aa0); pos: part-of-speech tags; pre: previously predicted NE tags; quo: flag signing that the word is between quotes; tri: trigger words.
Most systems used
- Maximum entropy modeling (5)
- Hidden-Markov models (4)
- Connectionists methods (4)

Near all systems used external resources, e.g., gazetteers

Best systems performed hybrid learning approach
Semi-supervised NE: idea

- Define manually only a small set of trusted seeds
- Training then only uses un-labeled data
- Initialize system by labeling the corpus with the seeds
- 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
Semi-supervised NE-learning: idea

- Trusted seeds
- NE Database
- Unlabeled corpus
- Annotator
- NE Candidate selection
- Labeled corpus
- Pattern learner
- Patterns
References for NEE

• Neumann (2007): web course page
  – http://www.dfki.de/%7Eneumann/meta-ner/SoftWareProject.html
• http://en.wikipedia.org/wiki/Named_entity_recognition
• Yangarber, Lin, Grishman, Coling 2002
• Lin, Yangarber, Grishman, ICML 2003