Modelling Natural Language with Finite Automata

Karin Haenelt

14. Theorietag der Fachgruppe „Automaten und Formale Sprachen“ der GI
28.09.2004
Overview

• Introduction
  – Application Domain
  – Natural Language Properties
  – Three Standard Levels of Modelling
  – Main Types of Transducers for Natural Language Modelling

• Modelling with Finite-State Automata
  – Lexical and Morphological Analysis
  – Syntactic Analysis
  – Semantic Analysis

• Summary
  – Modelling Devices
  – Linguistic Adequacy
  – Practical Usefulness
  – Complexity
  – Mass Data Processing
Advantages of Finite-State Devices

• efficiency
  – time
    • very fast
    • if deterministic or low-degree non-determinism
  – space:
    • compressed representations of data
    • = search structure (= hash function)

• system development and maintenance
  – modular design and automatic compilation of system components
  – high level specifications

• language modelling
  – uniform framework for modelling dictionaries and rules
Modelling Goal

• specification of a formal language that corresponds as closely as possible to a natural language

• ideally the formal system should
  – never undergenerate
    (i.e. accept or generate all the strings that characterise a natural language)
  – never overgenerate
    (i.e. not accept or generate any string which is not acceptable in a real language)

• realistically
  – natural languages are moving targets (productivity, variations)
  – approximations are achievable
The Commission promotes information on products in third countries.
Main Types of Transducers for Natural Language Processing

transducers with
- input label
- output label
- weights
- natural language
- interpretation
- variability of data,
- ranking of hypotheses
- n additional output strings
  at final states
- delayed output of ambiguities
- assignment of interpretations

general case: - non-deterministic transducers with $\varepsilon$-transitions
optimisation: - determinisation and minimisation

© Karin Haenelt 2004, Modelling Natural Language with Finite Automata
Are Natural Languages Regular Languages?

• On which level and to what extent are natural languages regular languages?

• On which level and to what extent are regular languages adequate for natural language interpretation?

• How can mapping ambiguities between input languages and output languages be handled most efficiently?
• How can sources of non-determinism be reduced?
• How can finite-state technology be combined with other methods in order to process information which cannot be expressed with regular languages?
**Lexical Analysis of Natural Language Texts**

<table>
<thead>
<tr>
<th>Lexical Analysis</th>
<th>Mapping to Output Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recogn. Input L</td>
<td>Mapping to Normalised Forms</td>
</tr>
<tr>
<td>Tokenisation</td>
<td>Morphological Analysis</td>
</tr>
<tr>
<td>Identification of Basic Units (Words)</td>
<td>Assignment of Categories</td>
</tr>
</tbody>
</table>

| 0 | t | 1 | h | 2 | i | 3 | n | 4 | k | 5 | ε | 6 |

- **category:** verb
- **features:**
  - tense: present
  - pers: 3
  - num: singular

© Karin Haenelt 2004, Modelling Natural Language with Finite Automata
Properties of Natural Language Words

- very large set
- word formation: concatenation with constraints
  - simple words
  - compound word
    - productive compounding and derivations
      - Drosselklappenpotentiometer
      - organise → organisation → organisational,
      - re-organise → re-organisation → re-organisational
      - be-wald-en → *be-feld-en
    - contiguous dependencies
      - go-es → *walk-es
      - un-expect-ed-ly → *un-elephant-ed-ly
    - discontiguous (long distance) dependencies
      - expect-s → *un-expect-s
      - mach-st → *ge-mach-st
Modelling of Natural Language Words

• Concatenation rules expressed in terms of
  – meaningful word components (including simple words) (morphs)
  – and their concatenation (morphotactics)

• Modelling approach
  – lexicalisation of sets of meaningful components (morph classes)
  – representation of these dictionaries with finite-state transducers
  – specification of concatenation
Modelling of Natural Language
Words: System Overview

- morpheme classes
- morphotactics
- phon./orthograph. alternation rules
- special expressions

- stem
  (book:book) <affix>
  (box:box) <affix>

- affix
  (ε:+noun +sg) <#>
  (s:+noun +pl) <#>

concatenation of morpheme classes

lexicon
compiler

lexical transducer
### Modelling of Natural Language Words: Cases

<table>
<thead>
<tr>
<th>Phenomena</th>
<th>Finite-State Transducers</th>
<th>other</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>deterministic</td>
<td>non-determin.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Morpheme Classes and Morphotactics</th>
<th>det &amp; minimal</th>
<th>deterministic minimisation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>standard case</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>special cases: ambiguities and sources of non-determinism</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>mapping ambiguities</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>input:out. languages</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>pattern overlapping</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>in input language</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>non-disjoint morpheme classes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>special cases: non-regular patterns</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>long-distance dependencies</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Phonological and Orthographical Alternation Rules</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Special Expressions</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **ambiguity must be preserved for later analysis**
- **very large search space (heavy backtracking)**
- **reduction of size, +flag check**
- **network copies, explosion in size**
- **high level specification of regular expressions**
- **special (non-morphologic) regular expressions**
Modelling of Words: Standard Case

| noun-stem | book | <noun-suffix> |
| work | <noun-suffix> |

| noun-suffix | (ε:+N +Sg) | # |
| (s:+N +Pl) | # |

Dictionary System: Morpheme Classes as target of continuation

Lexical Transducer
-deterministic
-minimal
- additional output at final states
Modelling of Words: Mapping Ambiguities between i:o-Language

Lexical Transducer
- deterministic
- delayed emission at final state
- minimal
Modelling of Words: Overlapping of Matching Patterns

Lexical Transducer
- non-deterministic with $\varepsilon$-transitions:
  - determinisation not possible
  - infinite delay of output due to cycle

Ambiguities
- cannot be resolved at lexical level
- must be preserved for later analysis steps
require non-deterministic traversal

© Karin Haenelt 2004, Modelling Natural Language with Finite Automata
Modelling of Words: Non-Disjoint Morpheme Classes

• problem:
  – multiplication of start sections
  – high degree of non-determinism

• solutions
  – pure finite-state devices
    • transducer with high degree of non-determinism
      – heavy backtracking / parallel search
      – slow processing
    • determinisation
      – explosion of network in size
  – extendend device: feature propagation along paths
    • merging of morpheme classes
    • reduction of degree of non-determinisms
    • minimalisation
    • additional checking of bit-vectors
Modelling of Words:
Non-Disjoint Morpheme Classes:
high degree of non-determinism

nou-n-stem
book  <noun-stem>
work  <noun-stem>

verb-stem
book  <verb-suffix>
work  <verb-suffix>

noun-suffix
(ε:+N +Sg) #
(s:+N +Pl) #

verb-suffix
(ε:+V) #
(ed:+V +past) #
(s:+V +3rd) #

problem:
- each of the subdivisions must be searched separately
- (determinization not feasible:
  - leads to explosion of network or
  - not possible)
Modelling of Words: Non-Disjoint Morpheme Classes: determinisation

deterministic version of the example

problem:
- sections multiplied
- explosion in size
- size of stem section very large
- example is only a tiny bit of a lexical transducer
Modelling of Words: Non-Disjoint Morpheme Classes: feature propagation

nou

nou

ver

sol

© Karin Haenelt 2004, Modelling Natural Language with Finite Automata
Modelling of Words: Long-Distance Dependencies

Constraints on the co-occurrence of morphs within words

<table>
<thead>
<tr>
<th>Contiguous Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>mach-e</td>
</tr>
<tr>
<td>mach-st</td>
</tr>
<tr>
<td>mach-t</td>
</tr>
<tr>
<td>mach-en</td>
</tr>
<tr>
<td>mach-t</td>
</tr>
<tr>
<td>mach-en</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>wachs-e</td>
</tr>
<tr>
<td>wach-e</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>invalid sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>*ge-mach-e</td>
</tr>
<tr>
<td>*ge-mach-st</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discontiguous Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-mach-t</td>
</tr>
<tr>
<td>ge-wachs-t</td>
</tr>
</tbody>
</table>

© Karin Haenelt 2004, Modelling Natural Language with Finite Automata
Modelling of Words: Long-Distance Dependencies

modelling alternatives

• pure finite-state transducer
  – copies of the network
  – can cause explosion in the size of the resulting transducer

• extended finite-state transducer
  – „context-free“ extension: simple memory: flags
  – special treatment by analysis and generation routine
  – keeps transducers small
Modelling of Words:
Long-Distance Dependencies: network copies

> 15,000 entries

problem:
can cause explosion in the size of the resulting transducer
Modelling of Words: Long-Distance Dependencies:

simple memory: flags

inflection suffixes

@require(-ge)

@set(+ge)

@require(+ge)

solution:
- state flags
- procedural interpretation (bit vector operations)

Beesley/Karttunen, 2003
Modelling of Words: Phon./Orth. Alternation Rules

- phenomena
  - pity → pittless
  - fly → flies
  - swim → swimming
  - delete → deleting
  - fox → foxes

- dictionary

<table>
<thead>
<tr>
<th>noun-stem</th>
<th>noun-suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>&lt;noun-suffix&gt; (ε:+N +Sg) #</td>
</tr>
<tr>
<td>fox</td>
<td>&lt;noun-suffix&gt; (s:+N +Pl) #</td>
</tr>
</tbody>
</table>

- rules: high level specification of regular expressions

  \[ E \rightarrow E \frac{x}{e/s}^{\text{a} \rightarrow \text{b}_c \quad [~[[~[?* b] a ?*] \quad [?* a \sim[c ?*]]] \quad \text{Beesley/Karttunen, 2003: 61} ]} \]
Modelling of Words: Phon./Orth. Alternation Rules

• compilation of lexical transducer:
  – construction of dictionary transducer
  – construction of rule transducer
  – composition of lexical transducer and rule transducer
Modelling of Words: Phon./Orth. Alternation Rules

e-insertion rule for English plural nouns ending with x,s,z („foxes“)

\[ e \rightarrow e / \left\{ \begin{array}{c} x \\ s \\ z \end{array} \right\} \]

© Karin Haenelt 2004, Modelling Natural Language with Finite Automata

Jurafsky/Martin, 2000, S. 78
Modelling of Words: Phon./Orth. Alternation Rules

- Dictionary
  - \( \Sigma \Delta \)

- Rule
  - \( \Delta \Omega \)

- Dictionary Rule
  - \( \Sigma \Omega \)
Modelling of Words: Special Expressions

- irregular mappings
  - did:do
- dates
  - 28.09.2004 / at the end of September / before Christmas
- number expressions (mapping includes reversing the sequence)
  - (siebenundsiebzig:77)

example: Sproat (2002:20)
Modelling of Words: Special Expressions

dictionaries of multiword lexemes – second level dictionaries?

• productive patterns:
  – third country, third party, third person, third world
  – but: third car, third house

• inflection
  – third country – third countries

• spelling variations
  – third country – third-country

• discontiguous components
  – third but associated countries
  – but: third and fourth person

• ambiguity
  – cannot be decided at the level of lexical analysis
  – must be represented for ensuing analysis steps

• modelling? duplicate entries? features? defer to later steps?
## Syntactic Analysis of Natural Language Texts

<table>
<thead>
<tr>
<th>Syntactic Analysis</th>
<th>Input Language</th>
<th>Output Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification of well-formed sequences of words</td>
<td>Assignment of Syntactic Categories</td>
<td>Assignment of Syntactic Structure</td>
</tr>
</tbody>
</table>

The good example

\[NP \quad NP \quad NP\]

- [NP the]
- [NP good]
- [NP example]

- NP the good example
Sequences of Words: Properties and Well-Formedness Conditions

• Syntactic Conditions
  – Concatenation
    • type-3 grammar
    • beyond type-3 grammar
    • beyond type-2 grammar

• Semantic Conditions
  – meaning of words
  – world knowledge (grouping of phrases)

• Textual Conditions
  – information organisation principles
Sequences of Words: Properties and Well-Formedness Conditions

syntactic conditions: type-3

regular language concatenations

• local word ordering principles
  – the good example
  – could have been done

• global word ordering principles
  – (we) (gave) (him) (the book)

* example good the
* been could done have

* (gave) (him) (the book) (we)
concatenations beyond regular languages

- centre embedding \((S \rightarrow a S b)\)

- obligatorily paired correspondences
  - \textit{either ... or, if ... then}
  - can be nested inside each other
Sequences of Words: Properties and Well-Formedness Conditions

syntactic conditions: beyond type-2

- concatenations beyond context-free languages
  - cross-serial dependencies

\[ \begin{array}{c}
  \text{Jan säit das mer d'chind em Hans es huus lönd hälfe aastriiche} \\
  x_1 \quad x_2 \quad x_3 \quad y_1 \quad y_2 \quad y_3
\end{array} \]

\[ \begin{array}{c}
  \text{John said that we the children-acc let} \\
  \text{Hans-dat help} \\
  \text{the house paint}
\end{array} \]
Sequences of Words: Properties and Well-Formedness Conditions

**semantic conditions**

- semantic compatibility of words
  - *they drink beer*
  - *they enter beer*
- case frames
  - *we read a book*
  - *we walk a book*
- meaning of words
  - *colourless green ideas sleep furiously*
Sequences of Words: Properties and Well-Formedness Conditions

mixed conditions

1. The products referred to in paragraph 1 of point A may be held for sale or put on the market only in glass bottles which:

(a) are closed with:
- a mushroom-shaped stopper made of cork or other material permitted to come into contact with foodstuffs, held in place by a fastening, covered, if necessary, by a cap and sheathed in foil completely covering the stopper and all or part of the neck of the bottle,
- any other suitable closure in the case of bottles with a nominal content not exceeding 0.20 litres, and

(b) bear labelling conforming to the provisions of this Regulation.

(COUNCIL REGULATION (EC) No 1493/1999 of 17 May 1999 on the common organisation of the market in wine, Annex VIII G.)
Sequences of Words: Output Language / Grammar Properties

• status
  – cannot be observed
  – have been invented by scholars

• functions
  1. specification of all allowable sequences of words
  2. assignment of interpretation structures
Syntactic Grammars

• complete parsing
  – goal: recover complete, exact parses of sentences
  – closed-world assumption
    • lexicon and grammar are complete
    • place all types of conditions into one grammar
    • seeking the globally best parse of the entire search space
  – problems
    • not robust
    • too slow for mass data processing

• partial parsing
  – goal: recover syntactic information efficiently and reliably from unrestricted text
  – sacrificing completeness and depth of analysis
  – open-world assumption
    • lexicon and grammar are incomplete
    • local decisions

© Karin Haenelt 2004, Modelling Natural Language with Finite Automata
Syntactic Grammars: Complete Sentence Structure?
Syntactic Grammars: Complete Sentence Parsing computational problem

• combinatorial explosion of readings

List the sales of products in 1973
List the sales of products produced in 1973
List the sales of products in 1973 with the products in 1972
List the sales of products produced in 1973 with the products produced in 1972

3 readings
10 readings
28 readings
455 readings

Bod, 1998: 2
„All Grammars Leak“*

*Edward Sapir, 1921

• Not possible to provide an exact and complete characterization
  – of all well-formed utterances
  – that cleanly divides them from all other sequences of words which are regarded as ill-formed utterances

• Rules are not completely ill-founded

• Somehow we need to make things looser, in accounting for the creativity of language use
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

\[ T_1 \circ T_2 \]

\[ T_2 \]

\[ T_1 \]
### Syntactic Structure:
#### Finite-State Cascades (Abney)

**Finite-State Cascade**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Transition</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_0$</td>
<td>$T_1$</td>
<td>DP N P DN NN V-tns Pron Aux V-ing</td>
</tr>
<tr>
<td>$L_1$</td>
<td>$T_2$</td>
<td>NP P NP VP</td>
</tr>
<tr>
<td>$L_2$</td>
<td>$T_3$</td>
<td>NP PP VP</td>
</tr>
<tr>
<td>$L_3$</td>
<td>$T_4$</td>
<td>S</td>
</tr>
</tbody>
</table>

**Regular-Expression Grammar**

\[
L_1: \begin{cases} 
NP \rightarrow D ? N^* N \\
VP \rightarrow V - tns \mid Aux V - ing 
\end{cases} \\
L_2: \{ PP \rightarrow P \ NP \} \\
L_3: \{ S \ PP^* NP \ PP^* VP \ PP^* \} 
\]
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 bist of syntactic structure
Syntactic Structure: Finite-State Cascades (Abney)

- each transduction is defined by a set of patterns
  - category
  - regular expression
- regular expression is translated into a finite-state automaton
- level transducer
  - union of pattern automata
  - deterministic recognizer
  - each final state is associated with a unique pattern
- heuristics
  - longest match (resolution of ambiguities)
- external control process
  - if the recognizer blocks without reaching a final state, a single input element is punted to the output and recognition resumes at the following word
Syntactic Structure: Finite-State Cascades (Abney)

- patterns: reliable indicators of bits of syntactic structure
- parsing
  - easy-first parsing (easy calls first)
  - proceeds by growing islands of certainty into larger and larger phrases
  - no systematic parse tree from bottom to top
  - recognition of recognizable structures
  - containment of ambiguity
    - prepositional phrases and the like are left unattached
    - noun-noun modifications not resolved
Syntactic Structure: Finite-State Cascades (Abney)

• **extended patterns**
  – include actions
  – after a phrase with pattern p has been recognised an internal transducer for pattern p is used
    • to flesh out the phrase with features and internal structure
    • insert brackets (non-deterministic, not left-to-right)

• **features represented as bit vectors**
  – unification by bit operations
  – phrases are not rejected in case of unification failures
Syntactic Structure: Finite-State Approximation of Phrase Structure Grammars

- Pereira/Wright 1997
- convert the LR(0) characteristic machine $M(G)$ of a CFG $G$ into a FSA for a superset of the language $L(G)$ defined by $G$
- unfold $M(G)$ into a larger machine whose states carry information about the possible shift-reduce stacks $\text{aof } R(G)$
- flatten a shift-reduce recognizer fro a grammar $G$ into a FSA by eliminating the stack and turning reduce-moves into $\varepsilon$-transitions
Syntactic Structure: Bounding of Centre Embedding

- Sproat, 2002
- observation: unbounded centre embedding
  - does not occur in language use
  - seems to be too complex for human mental capacities
- finite state modelling of bounded centre embedding

\[
S \rightarrow \text{the (man|dog) } S_1 (\text{bites|walks}) \\
S_1 \rightarrow \text{the (man|dog) } S_2 (\text{bites|walks}) \\
S_2 \rightarrow \text{the (man|dog) } (\text{bites|walks}) \\
S_1 \rightarrow \epsilon \\
S_2 \rightarrow \epsilon
\]
Modelling of Natural Language Word Sequences: Approaches

<table>
<thead>
<tr>
<th></th>
<th>Finite-State Cascades</th>
<th>Context-free and mildly Context-Sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>structure</strong></td>
<td>shallow</td>
<td>deep</td>
</tr>
<tr>
<td><strong>adequacy</strong></td>
<td>approximation</td>
<td>overgeneration</td>
</tr>
<tr>
<td><strong>decision</strong></td>
<td>clear identification of syntactic phenomena</td>
<td>no distinction of the nature of phenomena</td>
</tr>
<tr>
<td><strong>reliability</strong></td>
<td>islands of reliability</td>
<td>global optimisation</td>
</tr>
<tr>
<td><strong>robustness</strong></td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>identification of known pattern skipping of unknown patterns</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td>succeeds only if all patterns are known</td>
</tr>
<tr>
<td><strong>complexity</strong></td>
<td>( n \times</td>
<td>\text{levels}</td>
</tr>
<tr>
<td></td>
<td>example:</td>
<td>example:</td>
</tr>
<tr>
<td></td>
<td>100 words, 10 levels:</td>
<td>100 words, 10 rules/category (toy gram)</td>
</tr>
<tr>
<td></td>
<td>( O(1.000) )</td>
<td>( O(10.000.000) )</td>
</tr>
<tr>
<td><strong>speed</strong></td>
<td>fast</td>
<td>slow</td>
</tr>
</tbody>
</table>
### Modelling of Natural Language Word Sequences: Cases (1)

<table>
<thead>
<tr>
<th>Well-Formedness Conditions</th>
<th>Finite-State Cascades</th>
<th>Context-free and mildly Context-Sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Syntactic</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>type-3 grammar</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>local sequences (chunks)</td>
<td>lower level of cascade</td>
<td>rules with terminal symbols</td>
</tr>
<tr>
<td>global sequences (phrases)</td>
<td>higher level of cascade</td>
<td>rules with terminal and non-terminal symbols</td>
</tr>
<tr>
<td><strong>beyond type-3 grammar</strong></td>
<td><strong>approximation</strong></td>
<td><strong>modelling</strong></td>
</tr>
<tr>
<td>embeddings of phrases</td>
<td>fixed number of levels</td>
<td>variable number of levels</td>
</tr>
<tr>
<td>centre embedding</td>
<td>bounded centre embedding</td>
<td>centre embedding feature processing</td>
</tr>
<tr>
<td>obligatorily paired correspondences</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>beyond type-2 grammar</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cross-serial dependencies</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Modelling of Natural Language
Word Sequences: Cases (2)

<table>
<thead>
<tr>
<th>Well-Formedness Conditions</th>
<th>Finite-State Cascades</th>
<th>Context-free and mildly Context-Sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ambiguity of noun sequences</td>
<td>longest match heuristics</td>
<td>overgeneration of reading hypotheses</td>
</tr>
<tr>
<td>ambiguity of prepositional phrase attachment</td>
<td>containment of ambiguity, underspecification</td>
<td>overgeneration of reading hypotheses</td>
</tr>
<tr>
<td>semantic compatibility of words</td>
<td>(local grammars)</td>
<td>(static) lexicalisation of cooccurrence restrictions</td>
</tr>
<tr>
<td>case frames</td>
<td>procedural</td>
<td>feature structure</td>
</tr>
<tr>
<td>meaning of words</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Textual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>information organisation principles</td>
<td>selected patterns</td>
<td>-</td>
</tr>
</tbody>
</table>
Semantic Analysis
Selected Approaches (1)

• **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.
Semantic Analysis
Selected Approaches (2)

• **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
Selected Approaches (3)

- **message understanding**
  - 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)

Relationship: TIE-UP
Entities: Bridgestone Sports Co.
          a local concern
          a Japanese trading house

JV Company: -
Capitalization: -
Summary: Finite-State Devices

- finite state transducers (determinised, minimised)
  - input label
  - output label
  - weights
  - n additional output strings
- natural language
- interpretation
- variability of data,
- ranking of hypotheses
- assignment of interpretation
- delayed output of ambiguities

- constraint propagation / flags (memory)
- cascading of transducers for approximating context-free descriptions and semantics
  - longest match heuristics
  - procedural external control (punting of unknown input sequences)
- high-level specifications of regular expressions (dictionaries, rules)
Summary: Linguistic Adequacy

- **word formation**
  - essentially regular language
- **sentence formation**
  - reduced recognition capacity (approximations)
    - corresponds to language use rather than natural language system
  - flat interpretation structures
    - clearly separate syntactic constraints from other (semantic, textual) constraints
  - partial interpretation structures
    - clearly identify the contribution of syntactic structure in the interplay with other structuring principles
- **content**
  - suitable for restricted fact extraction
  - deep text understanding generally still poorly understood
Summary: Practical Usefulness

• not all natural language phenomena can be described with finite-state devices
• many actually occurring phenomena can be described with regular devices
• not all practical applications require a complete and deep processing of natural language
• partial solutions allow for the development of many useful applications
Summary: Complexity of Finite-State Transducers for NLP

- theoretically: computationally intractable (Barton/Berwick/Ristad, 1987)
- SAT-problem is unnatural:
  - natural language problems are bounded in size
    - input and output alphabets,
    - word length of linguistic words,
    - partiality of functions and relations
  - combinatorial possibilities are locally restricted.
- practically, natural language finite-state systems
  - do not involve complex search
  - are remarkably fast
  - can in many relevant cases be determinised and minimised
Summary: Mass Data Processing

• **Context-free devices**
  – run-time complexity: $|G| \times n^3$, $|G| >> n^3$
  – too slow for mass data processing

• **Finite-state devices**
  – run-time complexity: best case: linear (with low degree of non-determinism)
  – best suited for mass data processing
Final State

Thank you very much for your attention.

Introduction
Morphology
Syntax
Semantics
Summary

speech recognition
phonology
part-of-speech tagging
tokenising
morphology
dictionaries
rules

translation
text:speech
shallow parsing
head-modifier-pairs
fact extraction

speech:text

analysis synthesis transfer

© Karin Haenelt 2004, Modelling Natural Language with Finite Automata
References


© Karin Haenelt 2004, Modelling Natural Language with Finite Automata
References


