LREC 2008
Marrakech, Morocco
28th May 2008

Some Fine Points of Hybrid Natural Language Processing

Peter Adolphs, DFKI GmbH, Language Technology Lab, Berlin

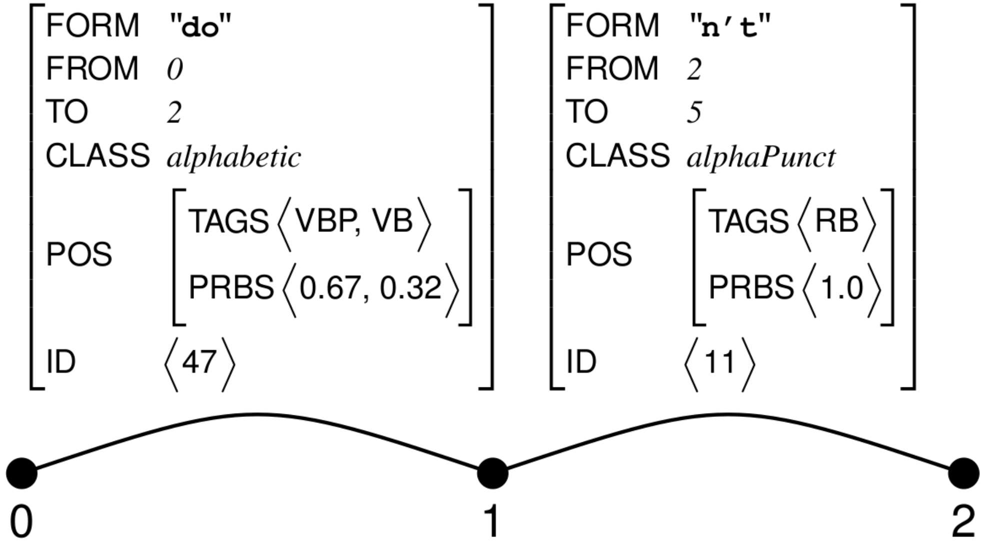
Stephan Oepen, Universitetet i Oslo, Department of Informatics
Ulrich Callmeier, acrolinx GmbH, Berlin
Berthold Crysmann, Universität Bonn
Dan Flickinger, Stanford University, CSLI
Bernd Kiefer, DFKI GmbH, Language Technology Lab, Saarbrücken

Motivation

- hybrid processing, integrating annotations of ‘shallow’ tools into HPSG parsing
- different tools make different assumptions
- example: PTB-style tokenizers for English
  - e.g.: Don't you! → <do, n't, you, !>
  - contracted verb forms are split
  - punctuation is split off the preceding word form
- we need to adapt annotations of different tools to the requirements of our grammar
- goal: a declarative, expressive, scalable device

Token Feature Structures

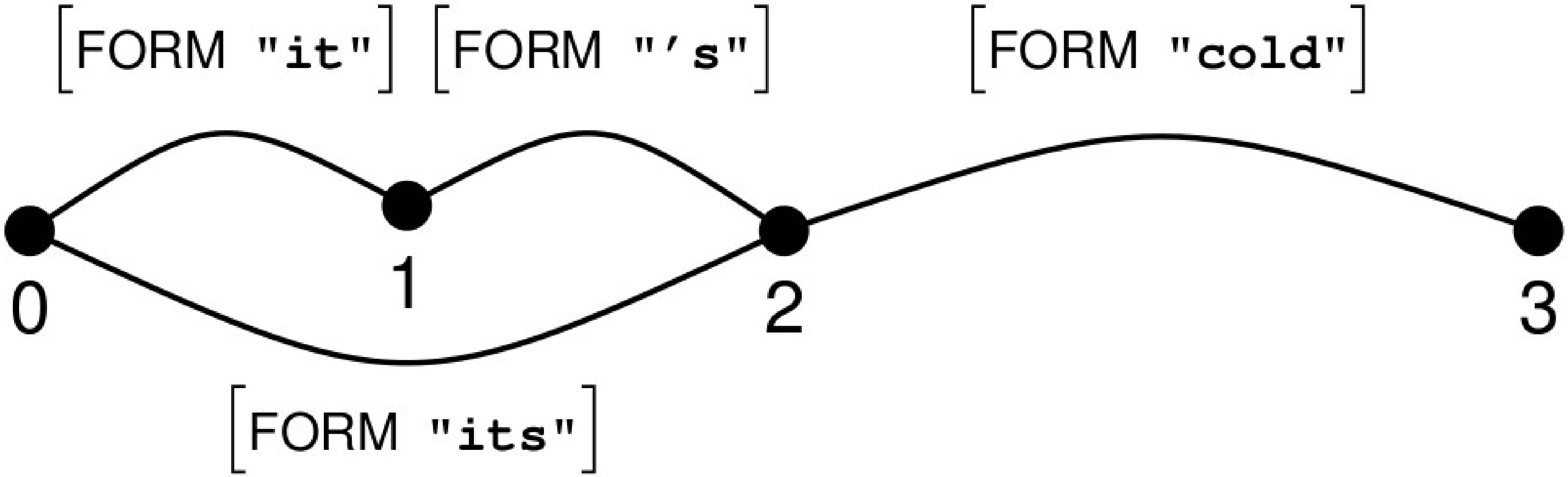
- feature structures for describing tokens
- different annotations provided as feature structures
- lattice of structured categories (token feature structures) as input to the parser



FORM "do"
FROM *0*
TO *2*
CLASS *alphabetic*
POS [TAGS ⟨VBP, VB⟩ PRBS ⟨0.67, 0.32⟩]
ID ⟨47⟩

FORM "n't"
FROM *2*
TO *5*
CLASS *alphaPunct*
POS [TAGS ⟨RB⟩ PRBS ⟨1.0⟩]
ID ⟨11⟩

Generalized Chart



- tools may assume different tokenization (paradigm case: input from speech recognizers)
- chart: dag whose vertices are abstract objects rather than indexed token boundary positions

Chart Mapping

- chart mapping: non-monotonic rewrite mechanism on feature structure chart edges
- general format:

[CONTEXT :] INPUT → OUTPUT

- CONTEXT, INPUT, OUTPUT are sequences of feature structures (each possibly empty)
- resource-sensitive: chart edges that let a rule fire may be removed (namely, all INPUT edges)

Chart Mapping – Example

$$\begin{bmatrix}\text{FORM} & \texttt{/\^{}(.+)\$/} \\ \text{TO} & \boxed{1}\end{bmatrix}, \begin{bmatrix}\text{FORM} & \texttt{"n't"} \\ \text{FROM} & \boxed{1}\end{bmatrix} \rightarrow \begin{bmatrix}\text{FORM} & \texttt{/\textbackslash 1n't/}\end{bmatrix}$$

- example: recombining split contracted forms
- rules extended with regular expression matches
- regex capture groups can be referred to in the output
- rules themselves described as feature structures, thus we can use re-entrancies

Chart Mapping – Examples

- light-weight named entity recognition

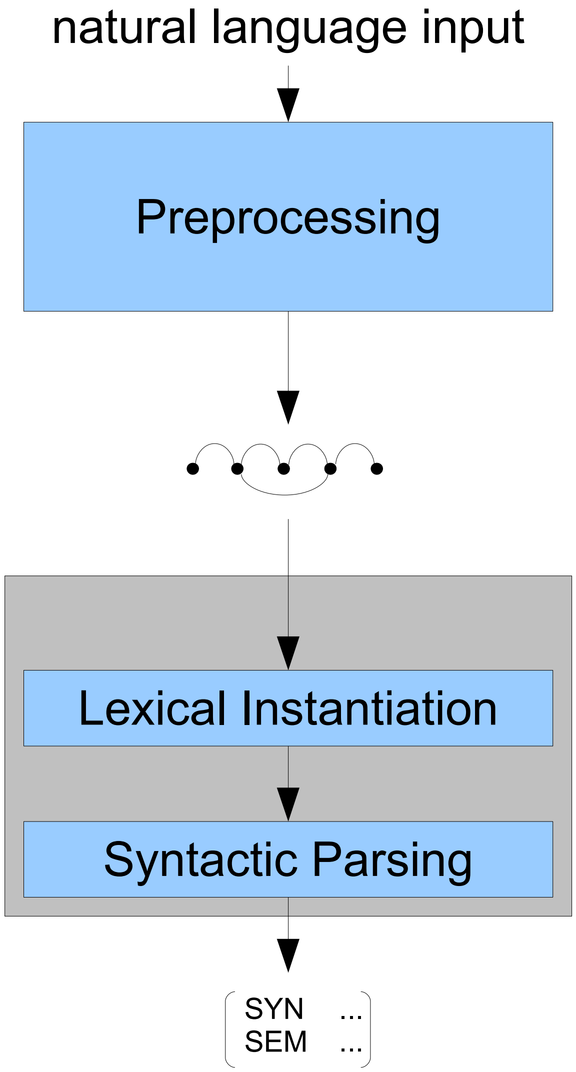
$$\left[\text{FORM}\ \ \texttt{/\textasciicircum([0-2]?[0-9]:[0-5][0-9])\$/}\right] \rightarrow \left[\begin{array}{ll}\text{FORM} & \texttt{/\textbackslash 1/}\\ \text{CLASS} & \mathit{clockTime}\end{array}\right]$$

- fixing broken tokenization

$$\left[\text{FORM}\ \ \texttt{/\textasciicircum(.+:)([a-zA-Z0-9].*)\$/}\right] \rightarrow \left[\text{FORM}\ \ \texttt{/\textbackslash 1/}\right], \left[\text{FORM}\ \ \texttt{/\textbackslash 2/}\right]$$

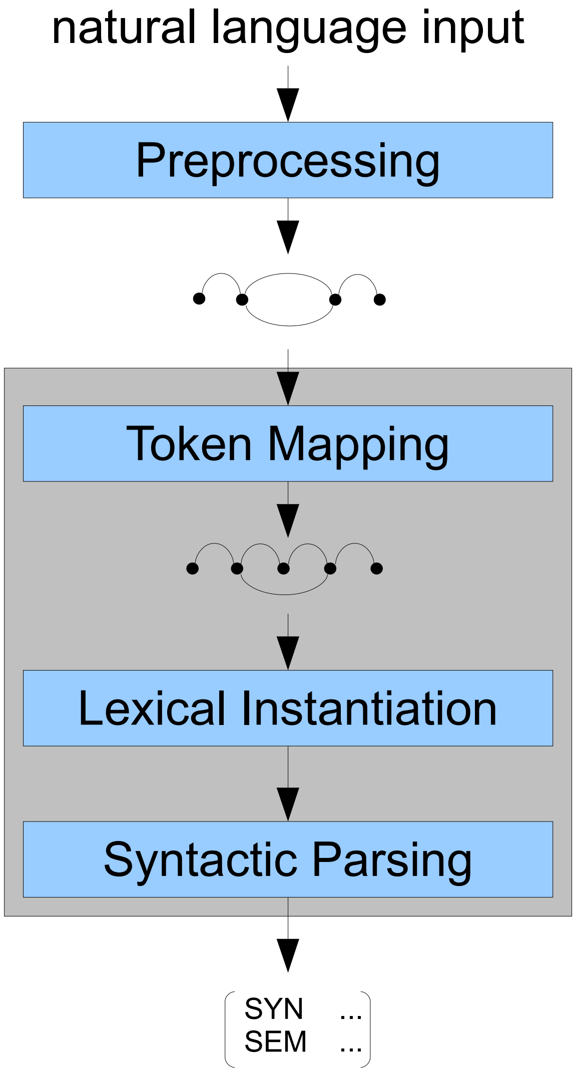
Previous Architecture (Simplified)

- preprocessing has to provide the input chart as expected by the grammar
- this has to be ensured by specialized conversion routines without recourse to the grammar
- changes to the grammar have to be reflected in these data adaptation routines



Proposed Architecture (Simplified)

- proposal: token mapping performs certain preprocessing steps within the grammar
- advantages:
  - full control for the grammar writer, using the same formalism as for the grammar
  - makes assumptions by the grammar explicit
  - removes complexity from preprocessing



Hybrid Processing

- shaping the search space of the parser:
  - widening search space (e.g. unknown word handling)
  - narrowing search space (e.g. removing / postponing the processing of edges)

- constraints on the search space
  - hard: categorial conditions for introduction / removal of chart edges
  - soft: probabilistic disambiguation, prioritize parser's tasks on the agenda

Lexical Instantiation

- native and generic lexical entries (les)
- selection of appropriate generic lexical entries originally controlled by the parser (hard-coded)
- strategy:
  - map from part-of-speech tags to generic les
  - instantiate generic le for highest ranked pos tag where no native le is available
- disadvantage:
  - not flexible enough (e.g. no chain of responsibility)
  - partial lexical coverage: *We'll* ***bus*** *to Paris.*

Lexical Instantiation

- proposal: try to instantiate **all** generic les for **all** tokens
- token feature structure is unified into a predefined path in the lexical entry
- selection of compatible tokens by constraints on the token feature structure
- example:

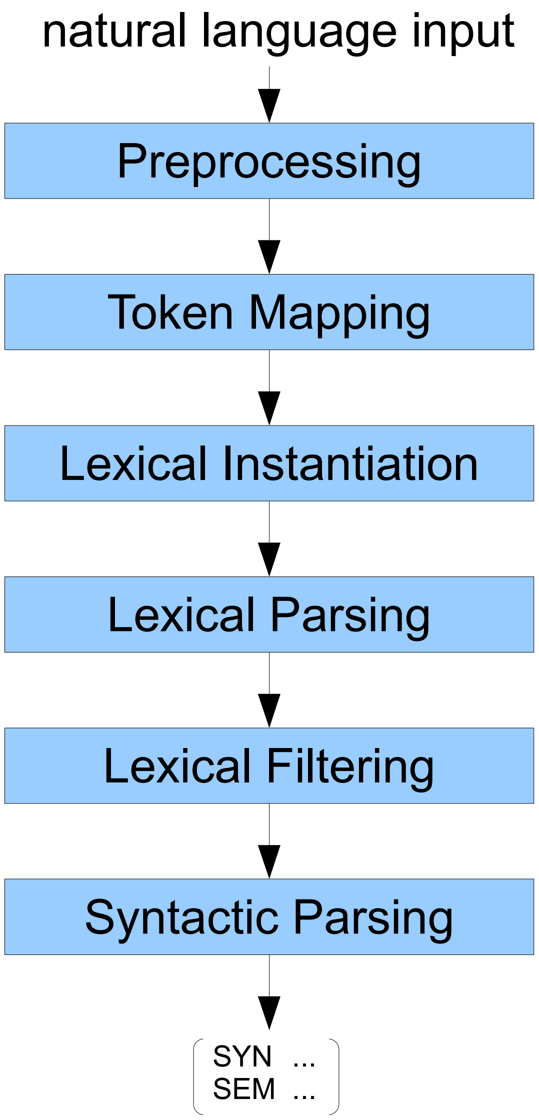
$$\text{genericname} := \begin{bmatrix} \text{SYNSEM} & \textit{noun\_nocomp\_synsem}, \\ \text{LOCAL.CAT.HEAD.MINORS.MIN} & \textit{named\_rel}, \\ \text{TOKENS} & \left\langle \begin{bmatrix} \text{POS.TAGS} & \langle \textit{NNP}, \ldots \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

Lexical Filtering

- after lexical instantiation, native and generic les may be available in the same chart cell
- we can restrict lexical instantiation by positing constraints on the token feature structures
- but we might also want to prevent some lexical chart edges in certain contexts (set operations)
- proposal: lexical filtering phase
- same formalism as for token mapping: chart mapping rules with empty OUTPUT list

Proposed Architecture

- use feature structures to describe tokens
- chart mapping: resource-sensitive rewriting of feature structure items
- chart mapping on token fs
- generic instantiation driven by compatibility with token fs
- lexical filtering with chart mapping



Applications

- fine grained control over instantiation of generic lexical entries
- mapping external morphological information into the grammar's universe
- chart dependency filter (optimizing parsing performance)
- activate syntactic rules only for certain spans of the input (e.g., in hybrid grammar checking)

Conclusions

- versatile device for many applications
- external information is made accessible to the grammar
- pre-processing can be better controlled with grammar-specific means
- reduces the need for special code inside and outside the parser
- outlook: consilidation of our current parsers and grammars

Thank you!

Acknowledgements

- DELPH-IN community and beyond, especially Nuria Bertomeu, Ann Copestake, Remy Sanouillet, Ulrich Schäfer and Benjamin Waldron for numerous in-depth discussions
- funding:
  - ProFIT program of the German federal state of Berlin and the EFRE program of the EU (to the DFKI project Checkpoint)
  - the University of Oslo (through its scientific partnership with CSLI)