Language Technology I:
Language Checking

Berthold Crysmann
crysmann@dfki.de

Overview

- **Spelling correction**
  - Application areas
  - Error types and frequency
  - Technology
    - Words & Non-words
    - Context-sensitive checking
- **Grammar checking**
  - Application areas
  - Error classification
  - Technology:
    - Constraint relaxation
    - Error anticipation
- **Controlled Language Checking**

Spelling correction - 1: Introduction

- **Application areas**
  - Authoring support
  - OCR
  - Preprocessing for IE, IR, QA, MT etc.
- **Typical error rates**
  - Typewritten text
    - 0.05% in edited newswire text
    - up to 38% in telephone directory lookups (Kukich 1992)
    - 1-3% in human typewritten text (Grudin 1983)
      cf. 1.5-2.5% in handwritten text (Kukich 1992)
  - OCR
    - 2-3% for handwritten input (Apple's NEWTON; Yaeger et al. 1998)
    - 0.2% for 1st generation typed input (Lopresti & Zhou 1997)
    - up to 20% for multiple copies/faxes (Lopresti & Zhou 1997)

Spelling correction - 2: Error types

- **Competence errors (cognitive)**
  - Ex.: **seperate* vs. *separate*
    **Lexikas* vs. *Lexika*
  - vary across speakers (learned, native, non-native)
  - Error reasons:
    - phonetic: see above
    - homonyms: *piece* vs. *peace*
- **Performance errors (typographic)**
  - Ex.: **speel* vs. *speel*
  - Single error misspellings account for 80% of non-words (Damerau 1964)
    - insertion: **ther* vs. *the*
    - deletion: **th* vs. *the*
    - substitution: **thw* vs. *the*
    - transposition: **hte* vs. *the*
  - Error reason (Grudin 1983):
    - substitution of adjacent keys (same row/column) and hands account for 83% of novice substitutions (experts: 51%)

Source: Berthold Crysmann 2005

Spelling correction - 2: Error types

- **OCR**
  - Ex. (Lopresti & Zhou 1997):
    The quick brown fox jumped over the lazy dog.
    'lhe q~ick brown foxjurnps ovcr tb l azy dog.
  - Error types:
    - Substitution: *ovcr*
    - Multisubstitution: *'lhe, tb*
    - Space deletion/insertion: f*oxjurnps, l azy*
    - Failures: *q~ick*

Spelling correction 2: Technology

- **Detecting non-words**
- **Naïve approach: dictionary lookup**
  - Limited to error detection
  - Problematic with languages featuring productive morphology
  - Early spell checkers (e.g. UNIX spell) permit (unconstrained) combination with affixes
    - massive overgeneration
  - Current spell checkers incorporate true morphology component
  - Lexicon size
    - Large lexicon: legitimate, rare words may mask common misspellings (Peterson 1986): *won't* vs. *wont*
      “hidden” single error mispellings: 10% for 50,000 word dictionary, 15% for 350,000
    - Damerau & Mays 1989 show that, in practice, large lexica improve spelling correction

Spelling correction 2: Technology – Bayesian approach

- **Noisy channel model (Jelinek 1970):**
  first application to spell checking by Kernighan et al. 1990

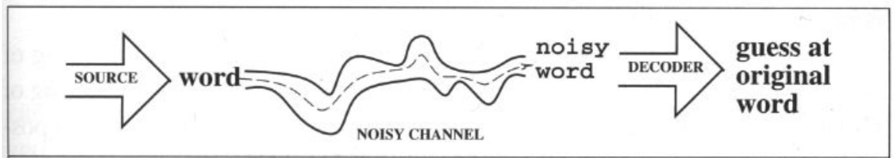


Figure 5.1 The noisy channel model.

- **Guess correct word based on observation of non-word:**
  ^w = argmax P(w|O), **w element of vocabulary V**
- **Equivalent to** ^w= argmax (P(O|w) P(w)) / P(O))
  (Bayesian rule)
- **Simplified to** ^w = argmax P(O|w) P(w)**, since P(O) constant**
  - Prior P(w) trivial to compute
  - Likelyhood P(O|w) must be estimated
- **Kernighan et al.'s checking algorithm:**
  - propose candidate corrections
  - rank candidates

Spelling correction 2: Technology – Bayesian approach

- **Candidate corrections**
  - Only single errors (insert,delete,transpose,substitute) considered by Kernighan et al.
- **Rank candidates**
  - ^c = argmax P(O|c) P(c)
  - P(c) equivalent to corpus frequency plus smoothing
  - P(O|c) estimated based on hand-annotated corpus of typos (Grudin (1983)
    - 4 confusion matrices (26x26) for letter insertion, deletion, transposition, substitution
  - Alternative (Kernighan et al. 1990)
    - EM-based estimation
    - Accuracy: 87% (best of 3)

Error	Correction	Transformation: Correct Letter	Transformation: Error Letter	Transformation: Position (Letter #)	Transformation: Type
acress	actress	t	–	2	deletion
acress	cress	–	a	0	insertion
acress	caress	ca	ac	0	transposition
acress	access	c	r	2	substitution
acress	across	o	e	3	substitution
acress	acres	–	2	5	insertion
acress	acres	–	2	4	insertion

Figure 5.2 Candidate corrections for the misspelling *acress*, together with the transformations that would have produced the error (after Kernighan et al. (1990)). "–" represents a null letter.

c	freq(c)	p(c)	p(t|c)	p(t|c)p(c)	%
actress	1343	.0000315	.000117	3.69×10^{-9}	**37%**
cress	0	.000000014	.00000144	2.02×10^{-14}	**0%**
caress	4	.0000001	.00000164	1.64×10^{-13}	**0%**
access	2280	.000058	.000000209	1.21×10^{-11}	**0%**
across	8436	.00019	.0000093	1.77×10^{-9}	**18%**
acres	2879	.000065	.0000321	2.09×10^{-9}	**21%**
acres	2879	.000065	.0000342	2.22×10^{-9}	**23%**

Figure 5.3 Computation of the ranking for each candidate correction. Note that the highest ranked word is not *actress* but *acres* (the two lines at the bottom of the table), since *acres* can be generated in two ways. The *del*[], *ins*[], *sub*[], and *trans*[] confusion matrices are given in full in Kernighan et al. (1990).

Spelling correction 2:
Technology – Multiple error correction

- **Minimal edit distance (Wagner & Fischer 1974):**
  - editing operations are insertion, deletion, substitution
- **Editing operations can be weighted**
  - Simplest weighting factor (all 1) also known as Levenshtein-distance)
- **Minimal edit distance can be combined with editing probabilities (product)**
- **Efficient integration with letter trees and FSAs possible (e.g. Wagner 1974, Mohri 1996, Oflazer 1996)**

- **Alternative: determine string distance based on shared n-grams**
  - Index lexicon entries according to string n-grams they contain
  - Maximise number of shared n-grams

Spelling correction 2:
Technology – Context-dependent error detection

- **Main objective: detect real-word errors**
  - Ex.: *piece – peace, it's – its, from – form*
- **Confusion sets (Ravin 1993)**
  - Group frequently confounded words into confusion sets
  - Develop heuristics to detect erroneous uses of elements within each set
- **n-grams**
  - Mays et al. 1991 employ 3-gram probabilities to compare sentences with their automatically generated variants
  - Mays et al. report correction rates of 70%
  - Combination of n-gram methods with predefined confusion sets (Golding & Schabes 1996) provides good results (98% corrections)
- **Other application:**
  - Errors in OCR of idiographs (e.g. Chinese) typically produce legitimate (though wrong) words
  - Hong 1996 employs bigram probabilities and CFGs to detect recognition errors and estimate the most likely word sequence

Grammar & style checking: Introduction

- **Application areas**
  - Authoring support
  - CALL (Computer-aided Language Learning)
  - Pre-editing for MT (see Controlled Language Checking)
- **Characterisation**
  - Ill-formed sentences/phrases derived from combination of well-formed words
  - May include detection of real-word spelling errors, in particular
  - Grammar checkers often include style checking rules
- **Style checking**
  - Document-internal consistency
  - Conformance to particular register

Grammar checking: Example errors 1 – Competence errors

- **Typical errors (German):**
  - Confusion of complementiser/relativiser
    - *Er schlug dem Kollegium vor, das*(s) montags und freitags keine Vorlesungen stattfinden*.
  - Comparatives
    - **größer ... wie* (dialectal)
  - Agreement
    - **ein großer(m) Fehlerkorpus(n)* (colloquial)
  - Blends
    - **meines Wissens nach*
- **Error type acquisition**
  - Error collections, prescriptive grammars (e.g. DUDEN), style & grammar guides (e.g. "Stolpersteine")
  - Corpus annotation

Grammar checking:
Example errors 2 – Performance errors

- **Typical errors**
  - Doublets
    - **the development of of a grammar checker*
    - **... denn Dubletten können auch nicht-lokal auftreten können*
  - Omissions
  - Transpositions
  - Typographically induced grammar errors
    - **eine besser Grammatiküberprüfung*
    - **a farmer form Oregon*
- **Error type acquisition**
  - Introspection
  - Corpus annotation

Source: Berthold Crysmann 2005

Grammar checking: Error classification – 1

- **3 dimensions (Rodríguez et al. 1996): source, cause, effect**
- **Source**
  - e.g. violation of particular grammatical constraints
  - language-specific
- **Cause**
  - Competence
  - Performance
    - Typographic errors
    - Editing errors
  - Input system (e.g. OCR)
- **Effect**
  - Word-level insertion, deletion, transposition, substitution
  - Constraint violation

Grammar checking:
Error classification 2 – Complexity

- **A 4th dimension: error detection/correction costs**
  - Grammatical modules:
    - Morphology
    - PoS-tagging
    - Chunk-parsing
    - Full parse
    - Sortal/Full semantics
    - Pragmatics
  - Locality of context
    - word
    - bounded context
    - sentence
- **Observation:**
  - Not always clear correspondence between error type and locality of context

Grammar checking: Error classification 2 – Complexity (example)

- **Example error:**
  - **meines Wissens nach*
  - Blend of *"meines Wissens(gen)"* with *"meinem(dat) Wissen(dat) nach"*
- **Highly frequent:**
  - 100 erroneous occurences in 8 million word corpus
  - 512 non-erroneous occurences
  - 16 occurences of alternate form (*"nach meinem Wissen"*)
  - 2 potential false positives (*"meines Wissens nach einem Proporz verteilt"*)
- **Complicating factors**
  - Ambiguity between pre- and postposition
  - Ambiguity between preposition and (stranded) verb particle

Grammar checking: Error classification 2 – Complexity (example)

- **Checking cost depends on linguistic context**
  - Clear true positive
    - Offending string immediately followed by finite verb
      **[meines Wissens nach] kam sie nie zu spät*
  - Almost certainly false positive
    - Offending string followed by dative NP (prepositional use of *"nach"*)
      [meines Wissens] [nach der Zerschlagung] des Faschismus eingeführt
  - Uncertain
    - Offending string at sentence boundary
      (*)*die Uhr ging meines Wissens nach* (separable verb prefix)
      **der Minister demissionierte meines Wissens nach*
    - Offending string followed by preposition
      **meines Wissens nach im Januar eingeführt*
      (*)*der Minister kam meines Wissens nach zum Essen* (PP-extraposition)

Grammar checking:
Error classification 2 – Complexity

- **Well-formed errors (Uszkoreit et al. 1997)**
- **Successful parse does not guarantee well-formedness**
  - **No friendship can lasts forever.* vs.
    No beer can lasts forever, even aluminum rots.
  - **Netscape showed a new browser a new browser at CeBIT.*
    I showed Mary the new boss at the party.
- **Large-scale grammars can often provide analyses for erroneous input**
  - by combining marked or infrequent constructions
    - **das Buch haben [der ø] [ø Schüler] gekauft*
    - combination of head-less NP, det-less NP with free dative
  - owing to absence of sortal restrictions and/or world knowledge

Grammar checking:
Error classification 3 – Performance vs. Competence

❑ One linguistic constraint is violated	❑ No direct correspondence with grammar
❑ There may be no correct alternative based on segment (e.g. missing lexical entry)	❑ A correct alternative always exists
❑ Checking for most error types should be optional (user customisable)	❑ No customisation necessary
❑ Simple error detection insufficient; explanation/correction needed	❑ Error detection sufficient
❑ Specialised modules according to native background and level of proficiency	❑ Special modules for specific input methods

Source: Berthold Crysmann 2005

Grammar checking:
Error classification 4 – Example error typology

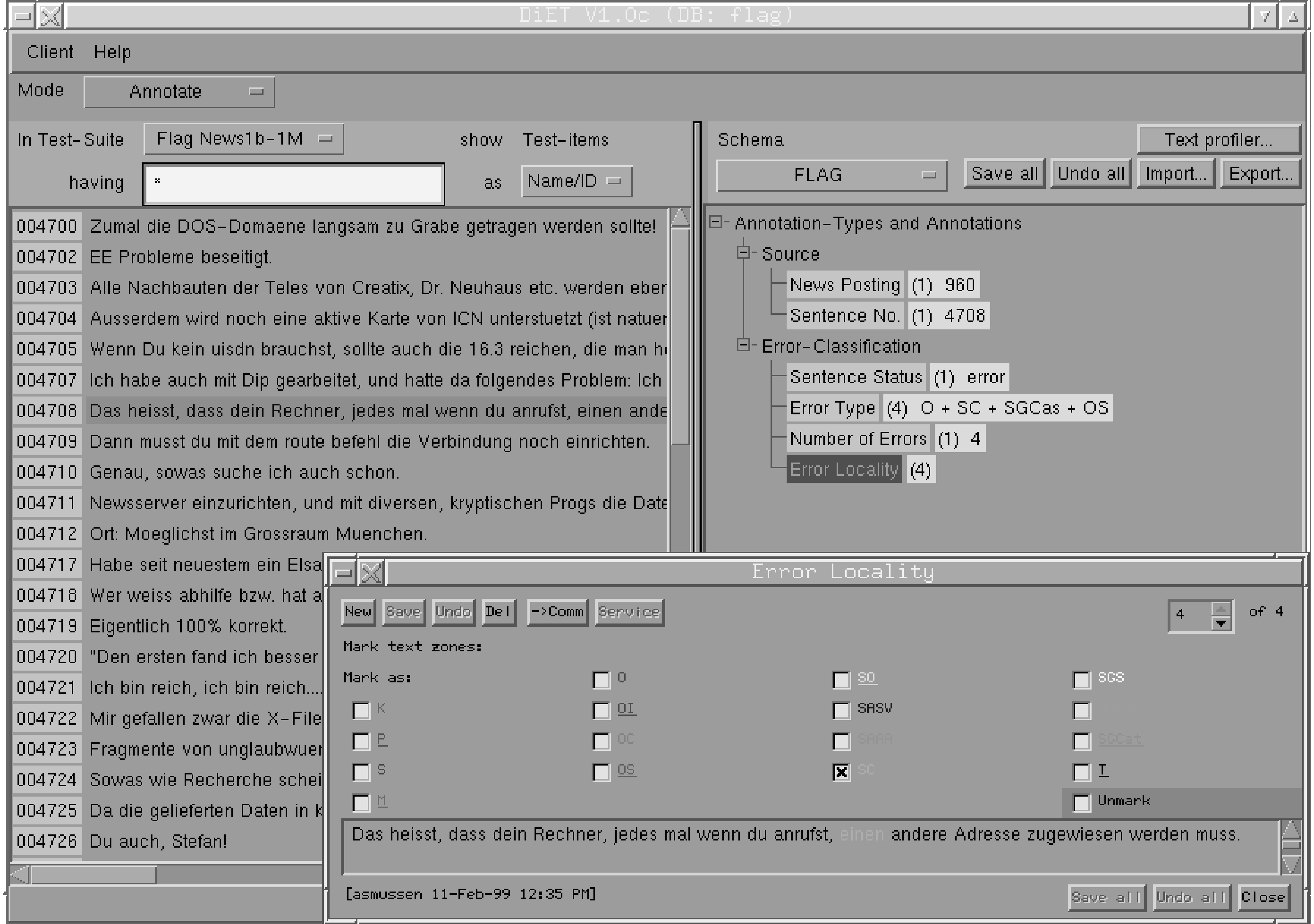
- **FLAG (Crysmann 1997; Becker et al . 2002)**
  - Hierarchical error classification
  - Annotation for
    - error type
    - error domain (NP)
    - error site (wrong adjectival form)
    - and lexical anchors (triggering condition for specific error types, e.g., neuter latinate nouns ending in *-us*)
  - Syntax errors:
    - Government (categorial, case, semantic selection etc.)
    - Concord (NP-internal)
    - Agreement (Subject-Verb, Antecedent-Anaphor)

Source: Berthold Crysmann 2005

Grammar checking:
Error classification 5 – Error frequency

- **Overall scarce distribution of grammatical errors**
  - Punctuation errors more frequent than the sum of all other grammar errors
  - Problem: low a priori probability for true errors implies low precision
- **Schmidt-Wigger (1998)**
  - 7,500 sentences (BMW-corpus) manually annotated
  -

Error type	*Error frequency*
Punctuation	238
Capitalisation	17
Separation	46
Agreement	44
Other (repetitions,omissions)	18


DiET V1.0c (DB: flag)
Client Help
Mode Annotate
In Test-Suite Flag News1b-1M show Test-items
having * as Name/ID
004700 Zumal die DOS-Domaene langsam zu Grabe getragen werden sollte!
004702 EE Probleme beseitigt.
004703 Alle Nachbauten der Teles von Creatix, Dr. Neuhaus etc. werden eber
004704 Ausserdem wird noch eine aktive Karte von ICN unterstuetzt (ist natuer
004705 Wenn Du kein uisdn brauchst, sollte auch die 16.3 reichen, die man h
004707 Ich habe auch mit Dip gearbeitet, und hatte da folgendes Problem: Ich
004708 Das heisst, dass dein Rechner, jedes mal wenn du anrufst, einen ande
004709 Dann musst du mit dem route befehl die Verbindung noch einrichten.
004710 Genau, sowas suche ich auch schon.
004711 Newsserver einzurichten, und mit diversen, kryptischen Progs die Date
004712 Ort: Moeglichst im Grossraum Muenchen.
004717 Habe seit neuestem ein Elsa
004718 Wer weiss abhilfe bzw. hat a
004719 Eigentlich 100% korrekt.
004720 "Den ersten fand ich besser
004721 Ich bin reich, ich bin reich....
004722 Mir gefallen zwar die X-File
004723 Fragmente von unglaubwuer
004724 Sowas wie Recherche schei
004725 Da die gelieferten Daten in k
004726 Du auch, Stefan!
Schema
Text profiler...
FLAG
Save all
Undo all
Import...
Export...
Annotation-Types and Annotations
Source
News Posting (1) 960
Sentence No. (1) 4708
Error-Classification
Sentence Status (1) error
Error Type (4) O + SC + SGCas + OS
Number of Errors (1) 4
Error Locality (4)
Error Locality
New Save Undo Del ->Comm Service
4 of 4
Mark text zones:
Mark as:
K
P
S
M
O
OI
OC
OS
SO
SASV
SC
SGS
I
Unmark
Das heisst, dass dein Rechner, jedes mal wenn du anrufst, einen andere Adresse zugewiesen werden muss.
[asmussen 11-Feb-99 12:35 PM]
Save all
Undo all
Close

Grammar checking:
Error classification 5 – Error frequency

- **Becker et al. (2002)**
  - 60,000 sentences (paper annotation) from USENET news groups
  - 14,492 sentences in machine-readable form (error db)
  - Dense distribution corpus-specific
    - chosen to reduce reading time/error
    - performance errors
  - Error distribution
    - Orthography: 83%
    - Grammar: 16%
  - Subcategorisation errors (9.4%)
    - mainly erroneous elisions (6.1%)
    - Confusion of *dass/das* (1.7%)
  - Other results
    - Error site with subject-verb agreement: Verb in 56 of 63 cases

Error type	*Label*	*Token*
Syntax (general)	S	3
Subject-verb agreement	SASV	63
Antecedent-anaphor agreement	SAAA	1
Concord (NP-internal agreement)	SC	180
Word order	SO	79
Valency (general)	SG	0
Subcategorisation	SGCat	854
Case assignment	SGCas	102
Semantic selection	SGS	265
Σ Syntax		1547
Morphology	M	91
Othography (general)	O	2893
Punctuation	OI	1701
Capital vs. small letters	OC	2776
One word vs. separate words	OS	1100
Σ Orthography		7561
All		9108

Table 1.2 Distribution of Error Types

Source: Berthold Crysmann 2005

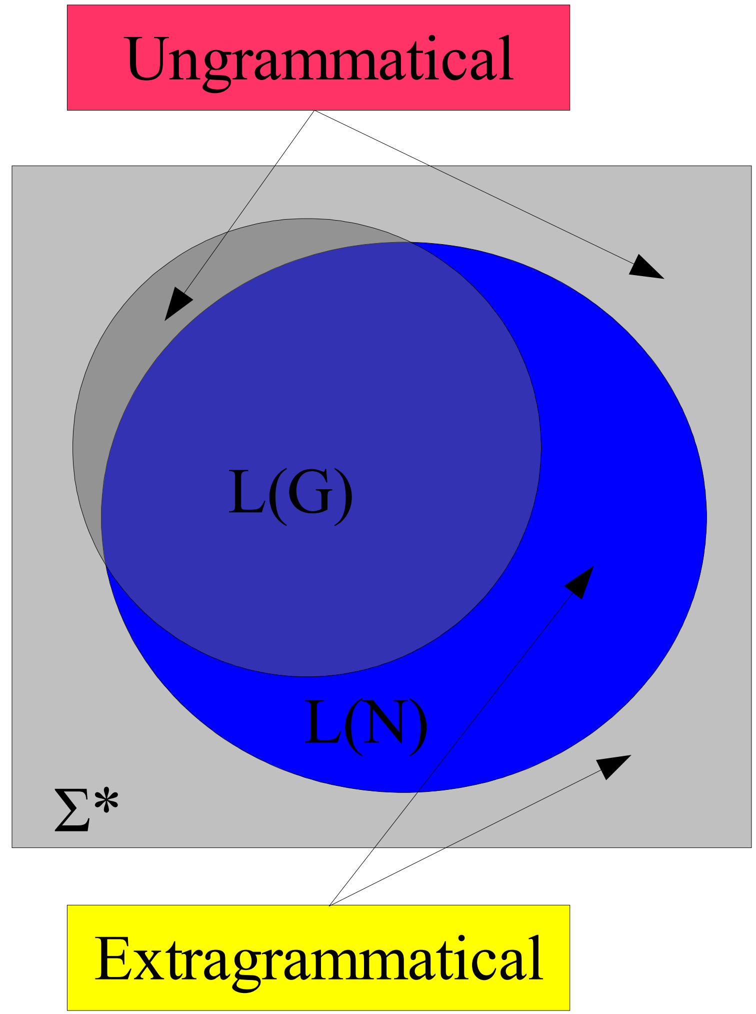
Grammar checking: Technology

- **Two paradigms:**
  - Parsing & Constraint relaxation
  - Error anticipation
- **Design criteria**
  - Speed
  - Error specification (positive vs. negative)
  - Error locality & correction
  - Feasibility

Source: Berthold Crysmann 2005

Grammar checking: Ungrammaticality and extra-grammaticality

- **Overgeneration and Undergeneration: L(G) ≠ L(N)**
  - Precision: Impeccable sentences erroneously flagged as erratic
  - Recall:
    - Implemented grammars may overgenerate
    - Syntactically, semantically or pragmatically marked constructions may mask true errors (well-formed errors)
- **Consequence: Importance of error models**
  - Manual construction (heuristics)
  - Automatic construction
    - complementation of FSAs (Sofkova 2000)
    - Negation of constraints (Menzel 1988)
  - Corpus-based



Source: Berthold Crysmann 2005

Grammar checking: Technology – Constraint relaxation

- **Robustness techniques (e.g., Stede 1992)**
  - Underspecification
  - Error anticipation
  - Constraint relaxation
  - Partial parsing (and fragment parsing)
- **Robustness in grammar checking**
  - Multiple pass strategy (e.g., CRITIQUE; Jensen et al. 1993)
    - Initial parse w/ full constraint set, relaxation on subsequent runs
    - Cost-neutral for well-formed input (L(G))
    - Partial results cannot be reused
  - Relaxable constraints (e.g., Douglas & Dale 1992 ; Rodríguez et al. 1996)
  - Parsing w/o constraints (Kudo 1988; Genthial et al. 1994)
    - Initial parse w/ CFG or DG backbone
    - Subsequent activation of morphosyntactic constraints (e.g., f-structure well-formedness constraints)
    - Word-order related errors (permutation, omissions etc.) undetectable

Grammar checking: Technology – Constraint relaxation 2

- **Robust PATR (Douglas & Dale 1992)**
  - Classify indvidual constraints as necessary/optional at different relaxation levels
  - On failure:
    - necessary constraint: proceed to next relaxation level
    - optional constraint: record failing constraint for error diagnosis
  - Assumption:
    - Errors are local
    - Error locality corresponds to constituency and parsing strategy

X0 → X1 X2

1	⟨X0 cat⟩	=	NP
2	⟨X1 cat⟩	=	Det
3	⟨X2 cat⟩	=	N
4	⟨X1 agr precedes⟩	=	⟨X2 agr begins⟩
5	⟨X1 agr num⟩	=	⟨X2 agr num⟩
6	⟨X0 agr num⟩	=	⟨X2 agr num⟩

Figure 3: Simple NP rule in the PATR formalism

Relaxation level 0:
necessary constraints = {1,2,3,4,5,6}
optional constraints = {}

Relaxation level 1:
necessary constraints = {1,2,3,6}
optional constraints = {5,4}

Figure 5: The relaxation specification for the NP rule, version 1: optional constraints

Relaxation level 1:
necessary constraints: {1,2,3}
relaxation packages:
(a) {5, 6}: Premodifier-noun number disagreement
(b) {4}: *a/an* error

Figure 6: The relaxation specification for the NP rule, version 2: grouped constraints

Grammar checking:
Technology – Constraint relaxation 3

- **Constraint relaxation in HPSG-style grammars (e.g. LateSlav)**
  - Relocate reentracies in HPSG-style rules to relational constraints
  - Assign diagnostic message to “error constraint”

$$\begin{bmatrix} \text{CAT} & \text{s} \\ \text{AGR} & \boxed{3} \\ \text{ERROR} & \boxed{4} \\ \text{DTRS} & \begin{bmatrix} \text{HD-DTR} & \begin{bmatrix} \text{CAT} & \text{v} \\ \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{COMP-DTR} & \begin{bmatrix} \text{CAT} & \text{n} \\ \text{AGR} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix} \wedge \text{agree}\left(\boxed{1},\boxed{2},\boxed{3},\boxed{4}\right)$$

```
agree(X,X,X,no_error).
agree(X,Y,X,agreement_error) :- X \= Y.
```

- **Alternative (e.g. JPSG)**
  - Generalise feature values on unification failure
  - Massive explosion of parse search space

Source: Berthold Crysmann 2005

Grammar checking: Technology – Constraint relaxation 4

- **Properties**
  - Implit incorporation of error model (relaxation technique/relaxable constraints)
- **Advantages**
  - Negative specification of error patterns (detect unforeseen errors)
  - Reuse of existing competence grammars
  - Validation of well-formed input (modulo well-formed errors)
- **Disadvantages**
  - Speed
    - Relaxation augments search space in parsing
    - Error sparseness (processing effort wasted on mostly correct sentences)
  - Error locality
  - Error diagnosis
  - Feasibility
    - Availability of large-scale high-precision grammars
    - Expressability of error patterns as constraints (e.g. omissions, insertions)
    - Integration of style rules (e.g. CRITIQUE sytem; Jenssen et al. 1993)

Source: Berthold Crysmann 2005

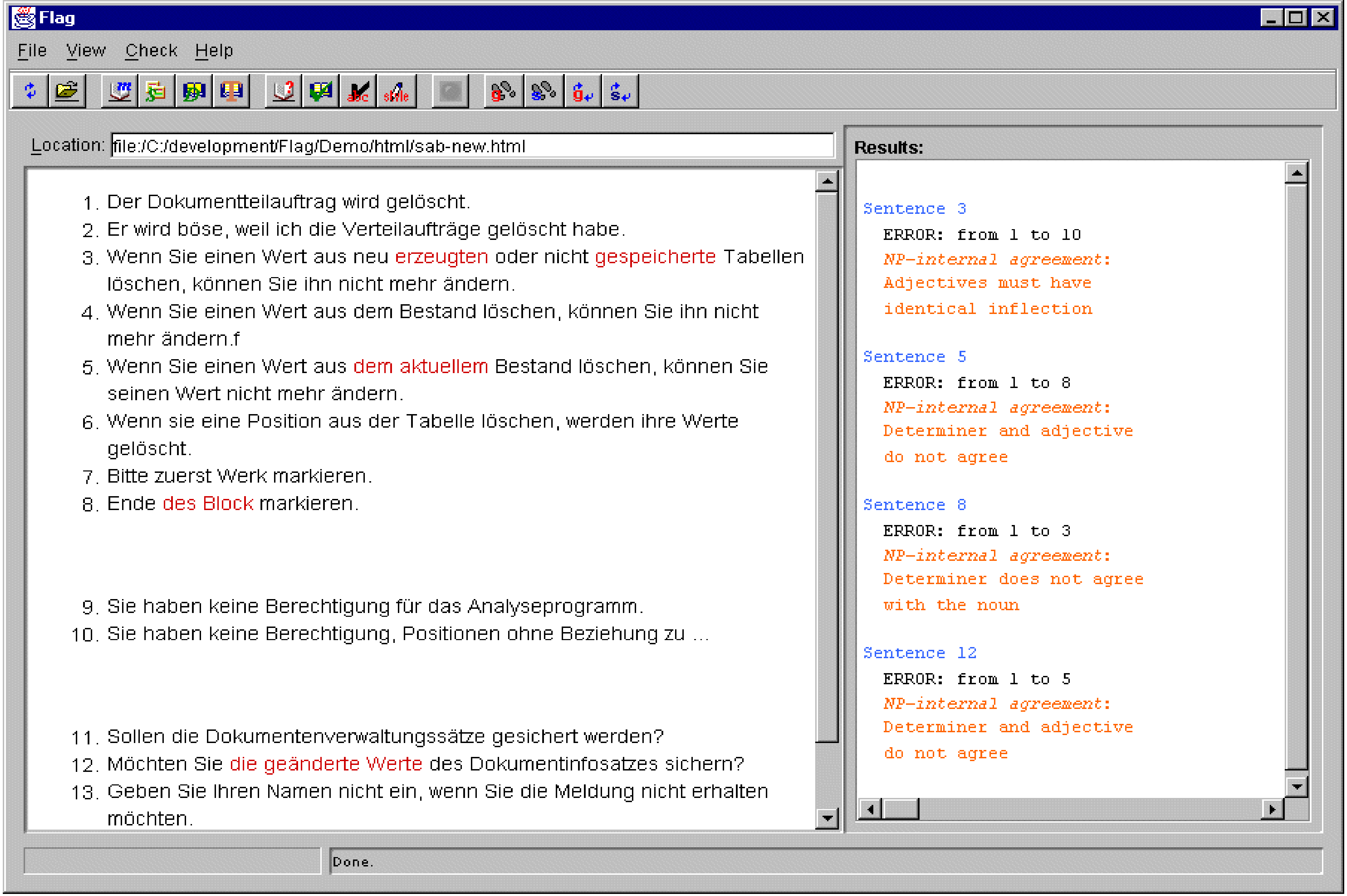
Grammar checking: Technology – Error anticipation 1

- **Properties**
  - Explicit error model
  - Pattern matching (heuristics)
- **Disadvantages**
  - Positive specification of error patterns (cannot detect unforeseen errors)
  - Only partial validation of well-formed input
- **Advantages**
  - Speed
  - Focussed processing & Resource adaptivity
  - Error locality
  - Detailed error diagnosis
  - Feasibility
    - Unavailability of large-scale high-precision grammars
    - Expressability of error patterns as constraints (e.g. omissions, insertions)

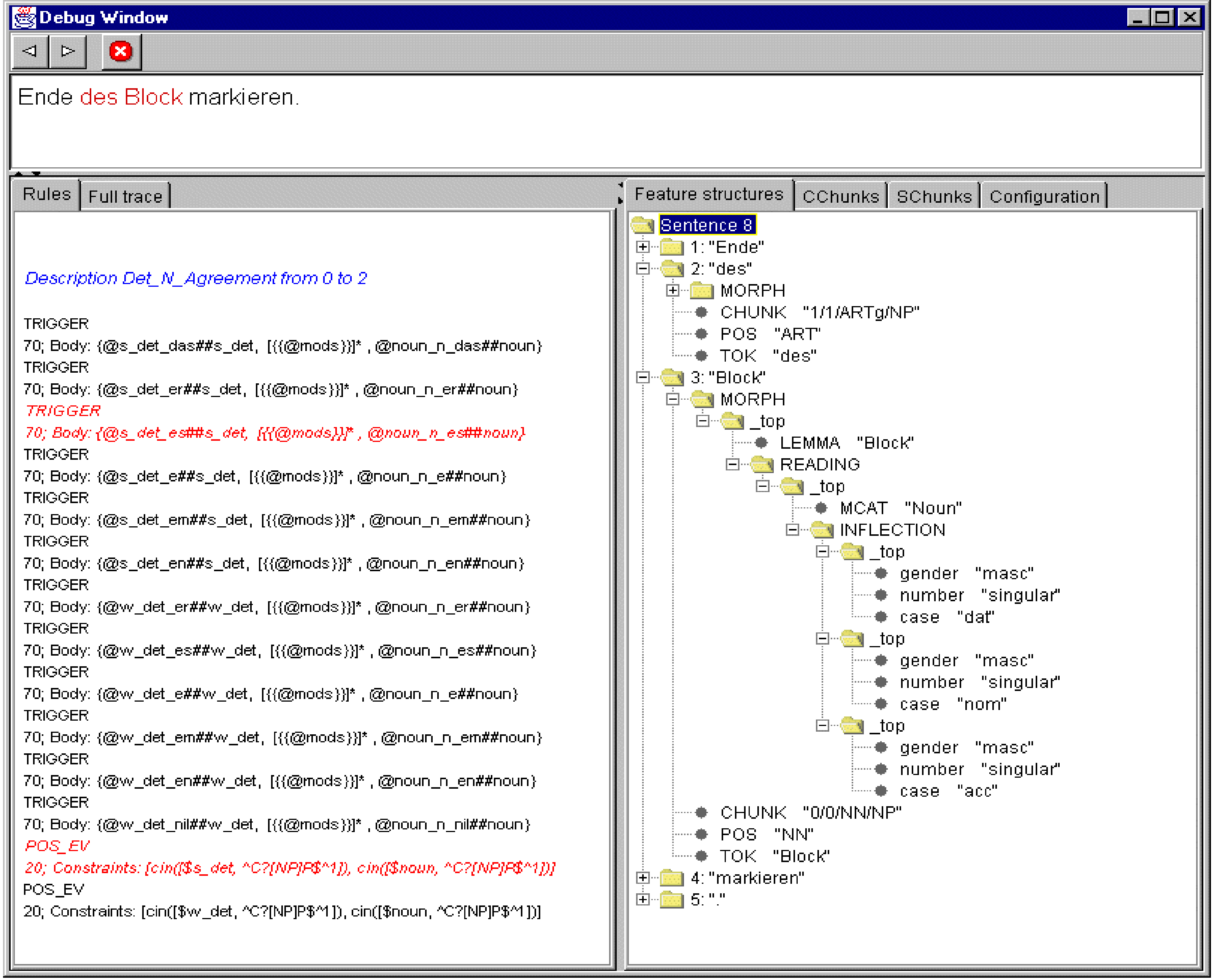
Source: Berthold Crysmann 2005

Grammar checking: Technology – Error anticipation 2

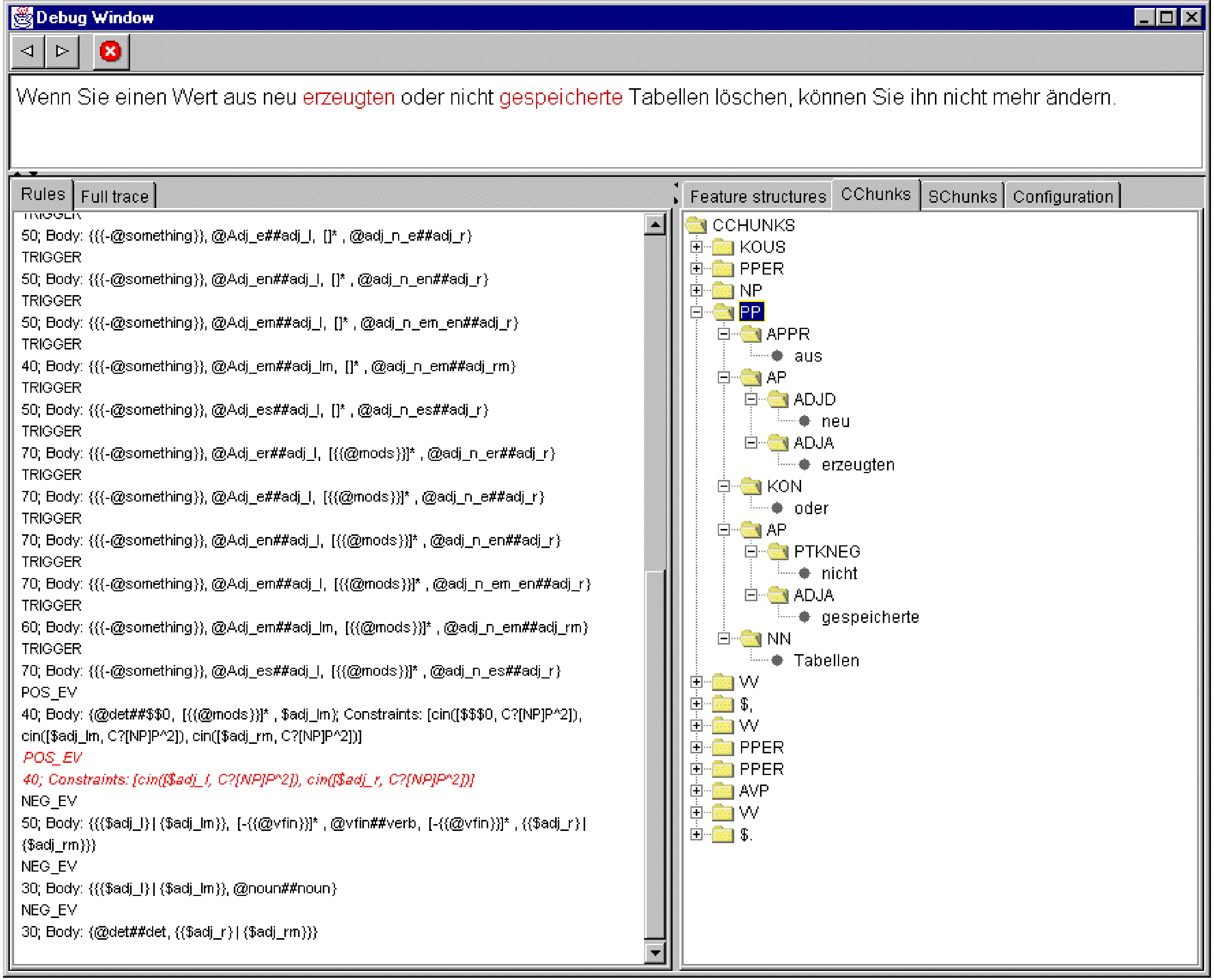
- **Example application: FLAG (Bredenkamp, Crysmann, Petrea 2000); now: acrocheck**
- **Linguistic annotation:**
  - Morphology (MULTEXT mmorph)
  - HMM PoS-Tagging (Brants 1999)
  - Chunk parsing (Skut & Brants 1998) & Topological parsing (Braun 1999)
- **Error detection**
  - Feature structure pattern matching (form, morphology, PoS)
  - Bottom-up integration of (partial) parsing
  - Systematic distinction between
    - initial trigger rules
    - confirming/disconfirming evidence (broader context, elaborate machinery)
  - Error heuristics (pattern matching rules) are weighted



Source: Berthold Crysmann 2005



Source: Berthold Crysmann 2005



Source: Berthold Crysmann 2005

Grammar checking: Summary & Outlook

- **Current status**
  - Low precision implies low user acceptance
  - Successful applications:
    - Non-native users
    - CALL
- **Perspectives**
  - Acquisition and integration of formal error models
  - Hybrid approaches
    - Deep/shallow processing
    - Error anticipation/relaxation

Controlled Language Checking: Introduction

- **Application areas**
  - Authoring support (technical documentation)
  - Pre-editing for MT
  - Information Management
- **Users**
  - Typically large, often multinational companies/organisations/industries
  - Factors:
    - short revision cycles
    - multiple source and target languages
    - separation between expert writers and non-expert translators
- **Goals**
  - Clarity
  - Consistency (including corporate style)
  - Translatability
    - elimination of ambiguous/difficult constructions, as well as jargon
    - homogeneity (for data-based MT and TM)

Controlled Language Checking: History

- **Caterpillar Functional English (in 1960s)**
- **Boeing Simplified English**
  - Aim: reduce complexity, ambiguity and vagueness
  - In-house development of checking technology (BSEC; production use since 1990)
  - Simplified English accepted as CL standard for entire industry: AECMA Simplified English
- **Other CL initiatives**
  - Automotive industry
    - General Motors (LANT)
    - Scania
    - BMW (IAI)
  - IT
    - SAP (DFKI/acrolinx)

Controlled Language Checking: Elements of a Controlled Language

- **Terminology**
  - Consistency
    - Approved/Unapproved variants
  - Patents (“Where do you want to go today?™”)
- **Style guides**
  - Complexity, e.g.
    - sentence length
    - nominal compounds
    - Active/Passive
    - Framing constructions (e.g. German separable particle verbs)
  - Ambiguity
    - PP-attachment
    - Word senses
  - Coherence
    - Correspondence between logical/temporal and surface order
  - Simplicity/Redundancy/Wordiness

Source: Berthold Crysmann 2005

Controlled Language Checking: Technologies

- **Terminology control**
  - Term bases
  - Morphological analysis (e.g. inflection, compounding)
- **Terminology mining**
  - TF/IDF
  - Term collocations
- **Word sense disambiguation**
  - one word – one meaning
  - Medical domain: *joint* (body part) vs. *joint* (#collective)
  - Airline domain:
    Round the edges of the round cap. If it then turns round and round as it circles round the casing, another round of tests is required. (Farrington 1996)

Controlled Language Checking: Technologies

- **Grammar checking (see above)**
- **Style checking**
  - Enforce adherance to sublanguage
  - CL-style rules often not formally defined
    - example-based
    - vague (Gricean)
    - proprietary
  - Styles make reference to
    - Document type:
User interface dialogues vs. manuals
    - Document structure:
Headings, bulleted lists
    - Relative position in document
  - Checking technology can only be complementary (Woicik & Hoard 1997)
    - address more mechanical aspects of a style guide
    - detect potential violations that may require human intervention

Controlled Language Checking: Technologies

- **Two approaches to style checking**
  - Grammar-based (e.g., BSEC, SECC)
  - Pattern-based (e.g., MultiLint, FLAG)
- **Comparison (Schmidt-Wigger 1998)**

Pattern-based	*Recall*	*Precision*
MultiLint (grammar)	57%	81%
MultiLint (style)	65%	92%
Grammar-based	*Recall*	*Precision*
BSEC (Wojcik 1990)	89%	79%
SECC (Adriaens 1994)	87%	93%

  - **Caution:**
    - Different corpora
    - Different rule sets