Core Technologies for IE

Günter Neumann & Feiyu Xu

{neumann, feiyu}@dfki.de

Language Technology-Lab

DFKI, Saarbrücken

Outline

- Overview
- Shallow technologies
- More NL understanding
- Hybrid approaches
- Additional Topics

Types of IE Tasks

Extraction of ...

- Topics
- Terms
- Named Entities
- Simple Relations
- Complex Relations (template filling)
- Answers to ad-hoc questions (QA systems)

- Some tasks require the merging of information from different parts of a document (e.g. template merging) and/or the fusion of information from several documents (information fusion).

IE Core Technologies

- A wide range of technologies for information extraction has emerged.

- Technologies range from non-linguistic approaches, e.g., methods for the exploitation of formatting and other formal properties of semi-structured documents all the way to truly linguistic approaches.

IE Core Technologies

- *Unstructured data* in the sense of computer science usually exhibit a complex linguistic structure. NLP methods are employed to detect and exploit this structure.

- Depending on the amount of structural analysis, NLP methods may be classified as *shallow* or *deep* methods.

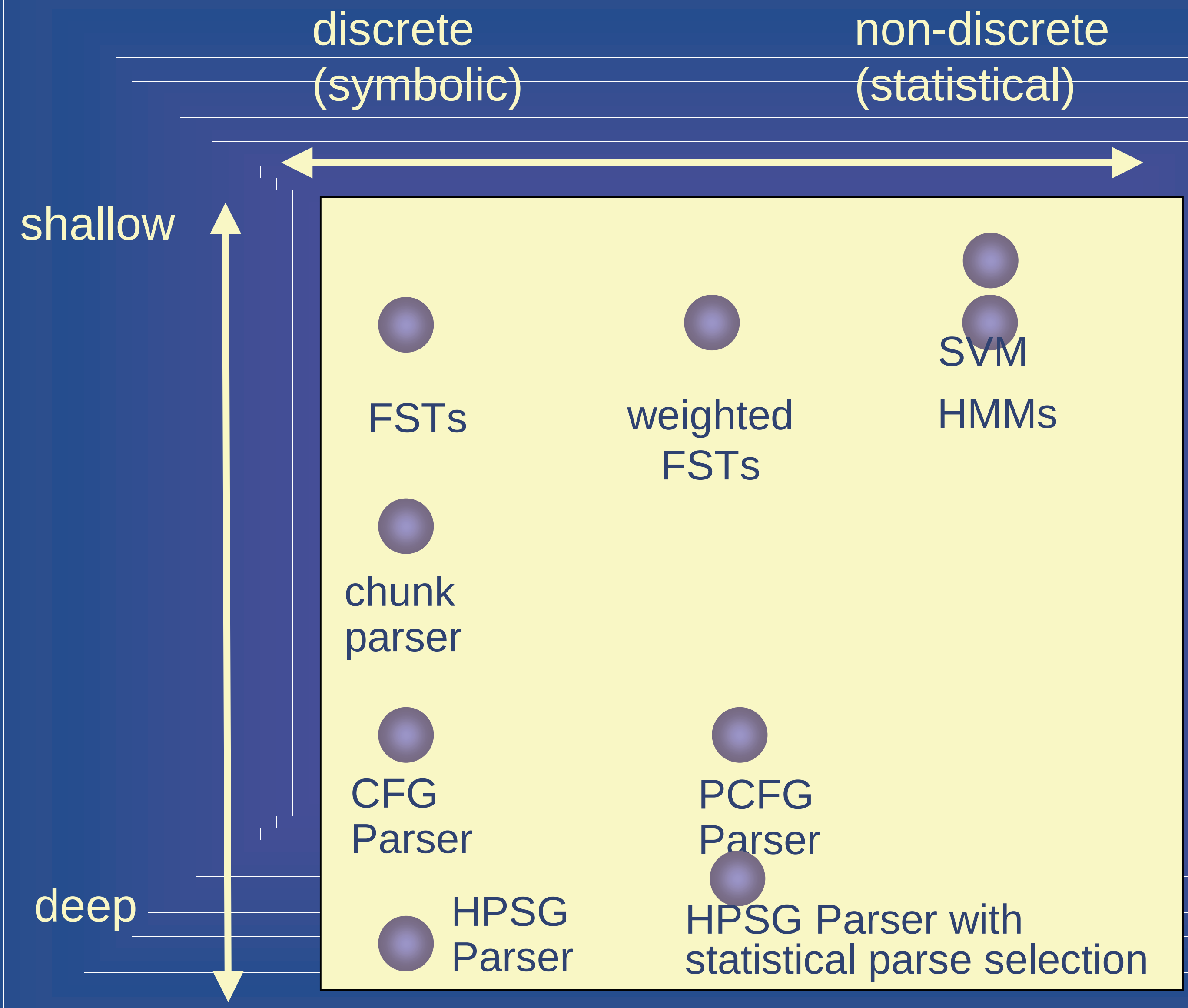
- Whereas deep methods try to analyze most or all of the linguistic structure, shallow methods derive much less structure.

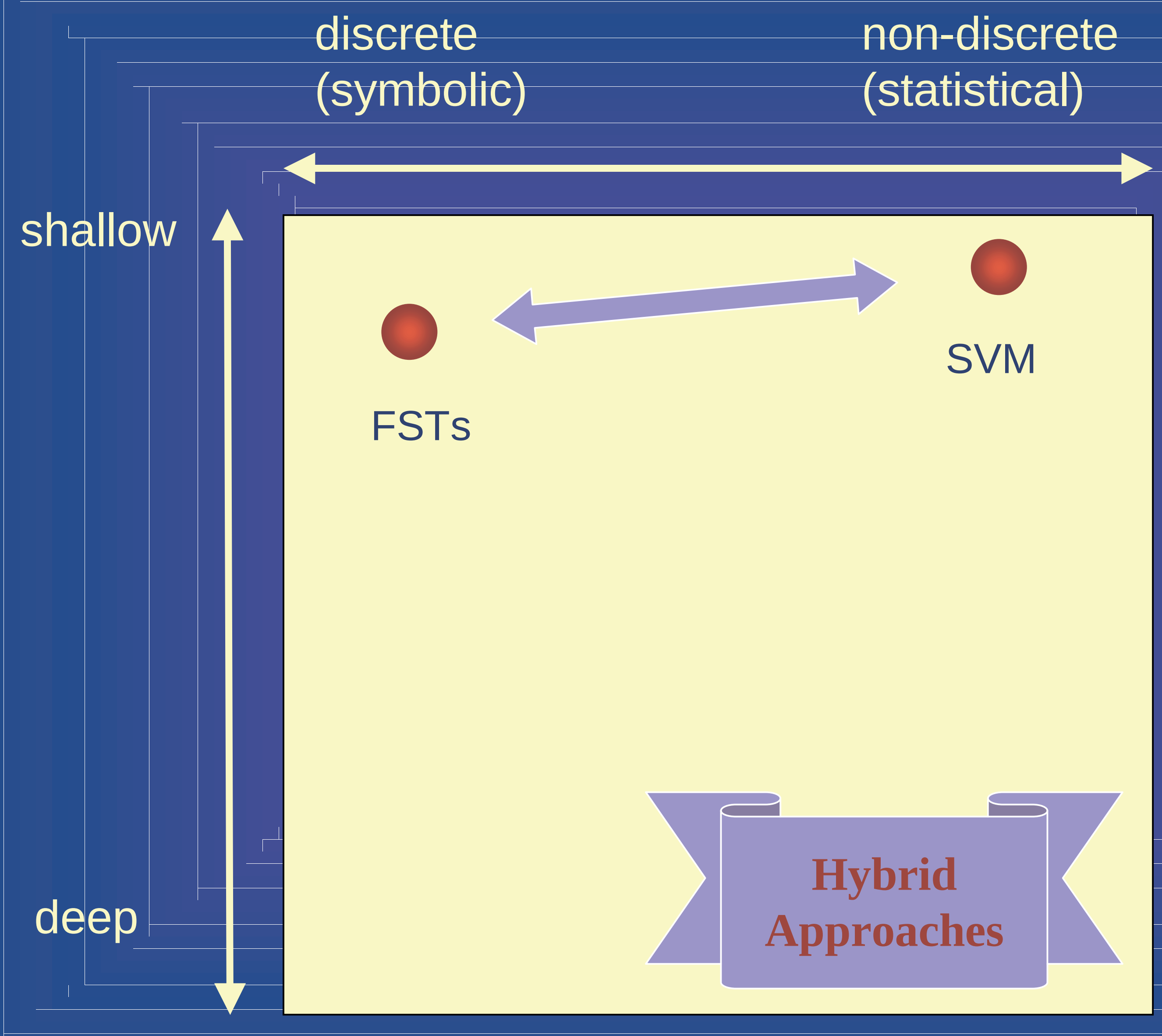
IE Core Technologies

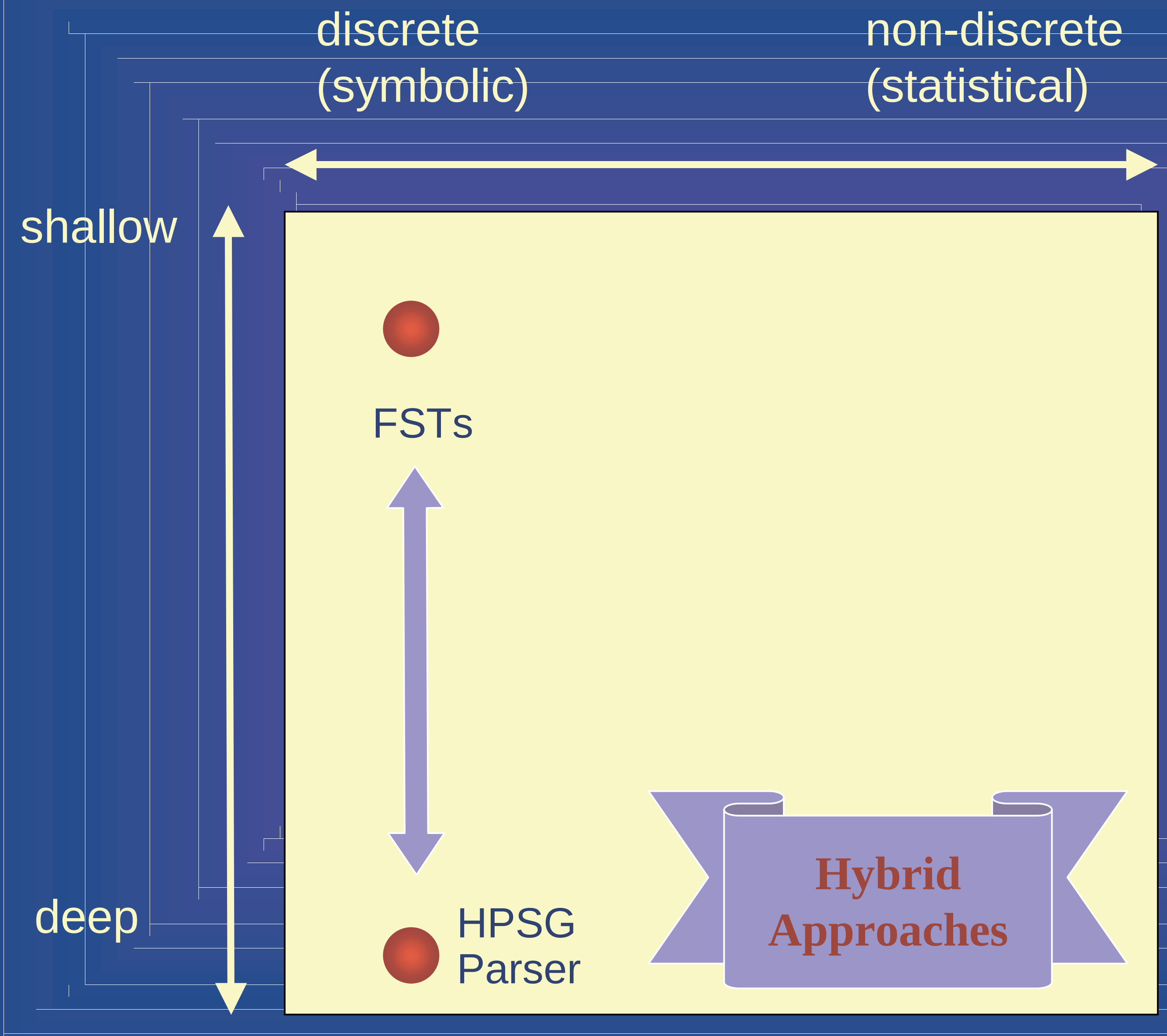
- Discrete methods are based on symbolic rules, patterns, principles such as rewriting rules or regular expressions.
- Nondiscrete methods are based on statistical/probabilistic techniques or neural networks.

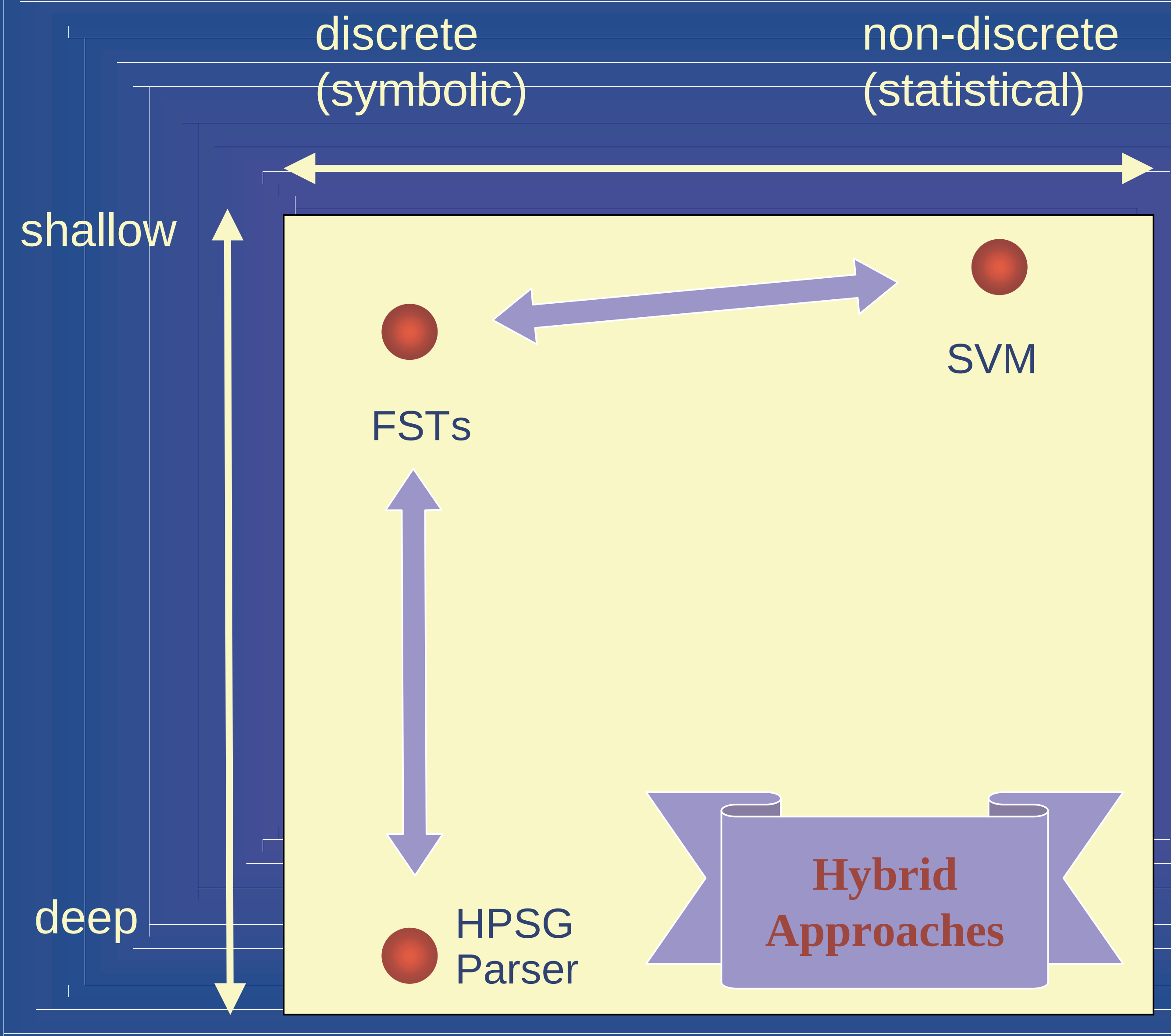
- The choice of the method depends on a number of criteria.
  - Task: what should be extracted? (topics, names, terms, relations, template fillers, answers)
  - What are the performance criteria of the application? (recall, precision, efficiency)
  - What are the design properties of the system: maintainability, adaptivity, interoperability, scalability, etc?

- Deep methods are accurate but lack efficiency and often also coverage (recall).
- Shallow methods are generally more efficient than deep ones and usually also often support wider coverage and therefore recall.
- Many discrete methods are well-suited for the manual design of rule or pattern sets.
- Non-discrete methods offer the advantage of modelling soft regularieties, especially the weighted contributions of several factors to classification decisions. For many non-discrete methods, very effective machine-learning schemes exist, supporting adaptivity and scalability.


discrete
(symbolic)
non-discrete
(statistical)
shallow
deep
FSTs
weighted
FSTs
SVM
HMMs
chunk
parser
CFG
Parser
PCFG
Parser
HPSG
Parser
HPSG Parser with
statistical parse selection


discrete
(symbolic)
non-discrete
(statistical)
shallow
deep
FSTs
SVM
Hybrid
Approaches


discrete
(symbolic)
non-discrete
(statistical)
shallow
deep
FSTs
HPSG
Parser
Hybrid
Approaches


discrete
(symbolic)
non-discrete
(statistical)
shallow
deep
FSTs
SVM
HPSG
Parser
Hybrid
Approaches

Shallow NLP

- Shallow NLP
  - Coarse-grained linguistic analysis tailored to a specific domain
  - Less structure
  - Direct mapping from linguistic to domain structure

- Widely used approach
  - Bottom-up, chunk recognition (named entities, NP, verb groups)
  - Template filling on basis of domain-specific verb-frames or other patterns

Bridgestone Sports Co. said Friday it had set up a joint venture in Taiwan with a Japanese trading house to produce golf clubs to be supplied to Japan.

Recognized as a joint venture event by applying template pattern

[COMPANY][SET-UP][JOINT-VENTURE] (others)* with[COMPANY]

Shallow Techniques

- Use of finite state transducers which map from surface form to IE relevant concepts and relations

- Systems and platforms
  - SRI Technologies
    - FASTUS
  - Sheffield
    - GATE
  - DFKI IE core technologies
    - IE from real-world German free texts
    - SPROUT

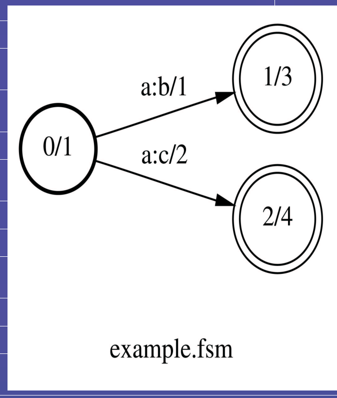
Why Finite State Technology?

- most of the relevant local phenomena can be easily expressed as finite-state devices
- time & space efficiency
- existence of efficient determinization, minimization and optimization transformations
- there exist much more powerful formalisms like context free grammars or unification grammars, but application developers prefer more pragmatic solutions

DFKI Finite-State Tools Overview

- efficiency-oriented implementation
- architecture and functionality is mainly based on the tools developed by AT&T
- representation: textual format vs. compressed binary format

```
0    1.0
----------------------
0    1    a    b    1.0
0    2    a    c    2.0
----------------------
1    3.0
2    4.0
```



- most of the provided operations are based on recent approaches (Mohri, Pereira, Roche, Schabes)

DFKI Finite-State Tools Overview

- operations are divided into four main pools:

<u>converting operations:</u>

- converting textual representation into binary format and vice versa
- creating graph representation for FSMs, etc.

<u>rational and combination operations:</u>

- union	- concatenation	- closure	- local extension
- reversion	- intersection	- inversion	- composition

<u>equivalence transformations:</u>

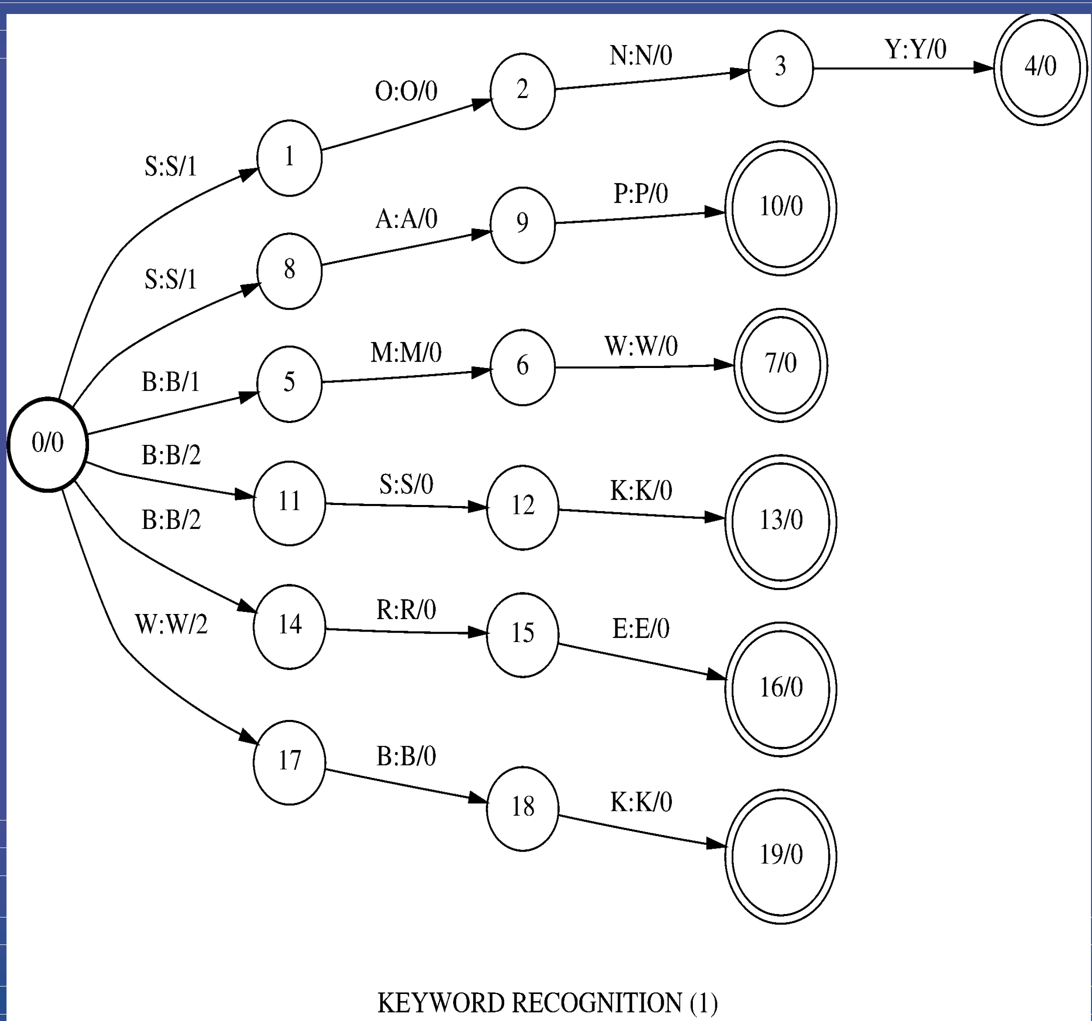
- determinization	- bunch of minimization algorithms
- epsilon removal	- trimming

<u>other:</u>

- extending input/output alphabet	- determinicity test
- collecting arcs with identical labels	- information about an FSM

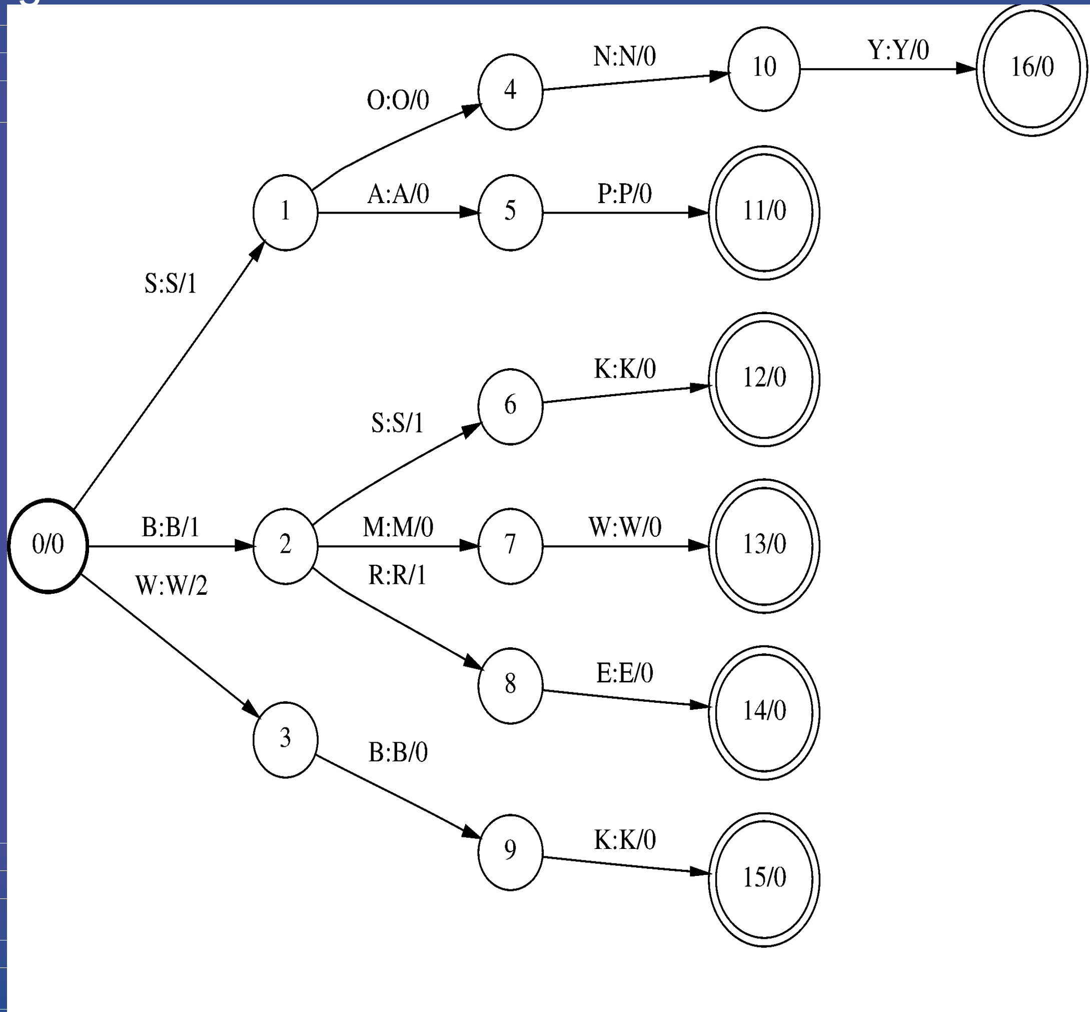
DFKI Finite-State Tools Examples

- Keywords Recognition - Automata Search



DFKI Finite-State Tools Examples

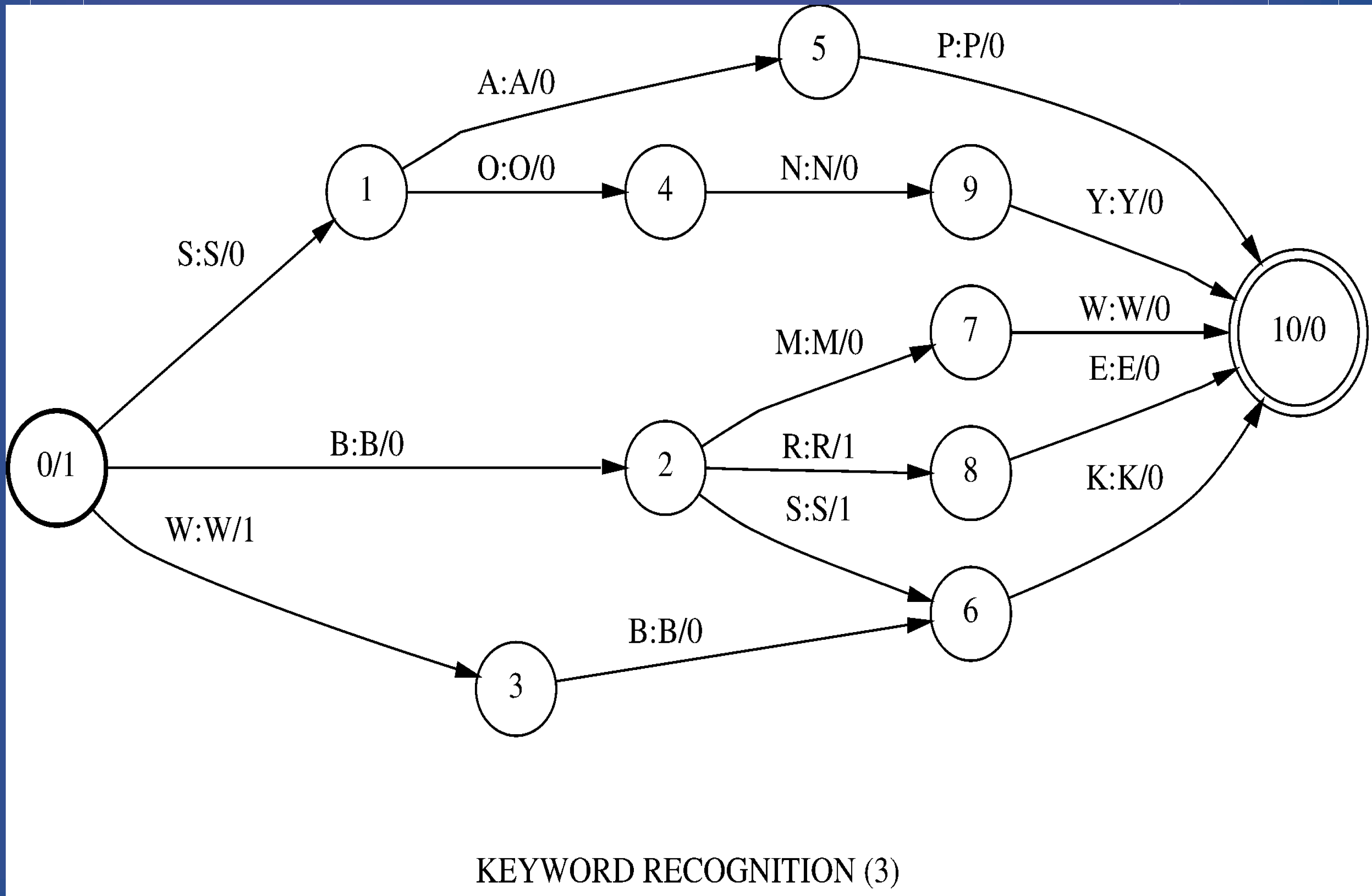
- Keywords Recognition - Deterministic Search



KEYWORD RECOGNITION (2)

DFKI Finite-State Tools Examples

- Keywords Recognition - Minimal Deterministic Search

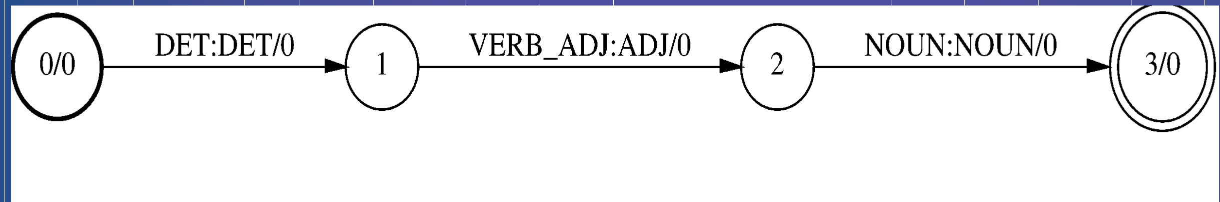


DFKI Finite-State Tools Examples

- contextual PART-of-SPEECH filtering rule:

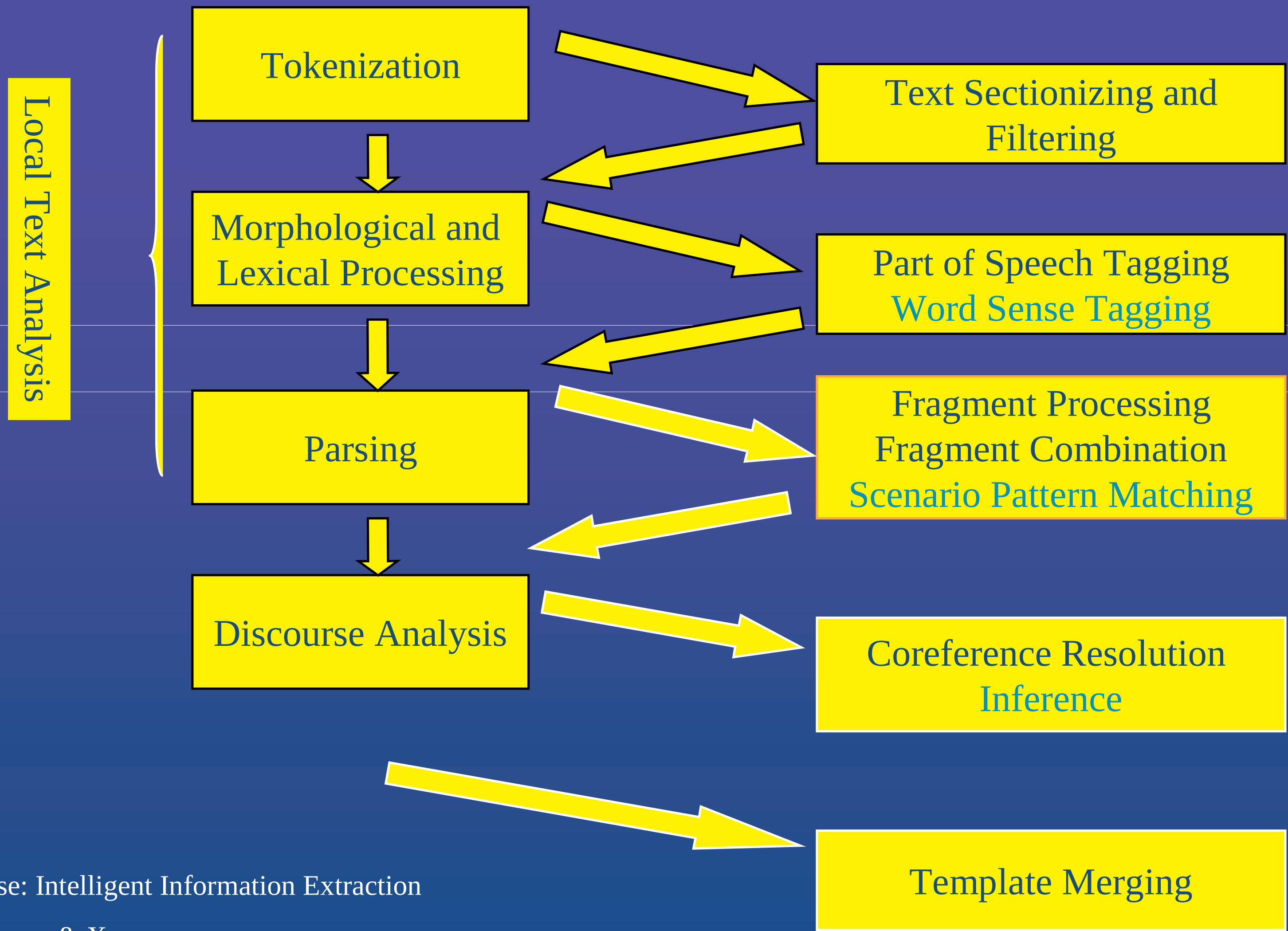
if the previous word form is a **determiner** and the next word form is a **noun** then
filter out the **verb reading**

example: ... die bekannten Bilder .... („*the known pictures*“)



POS-FILTERING (1)

Traditional IE Architecture

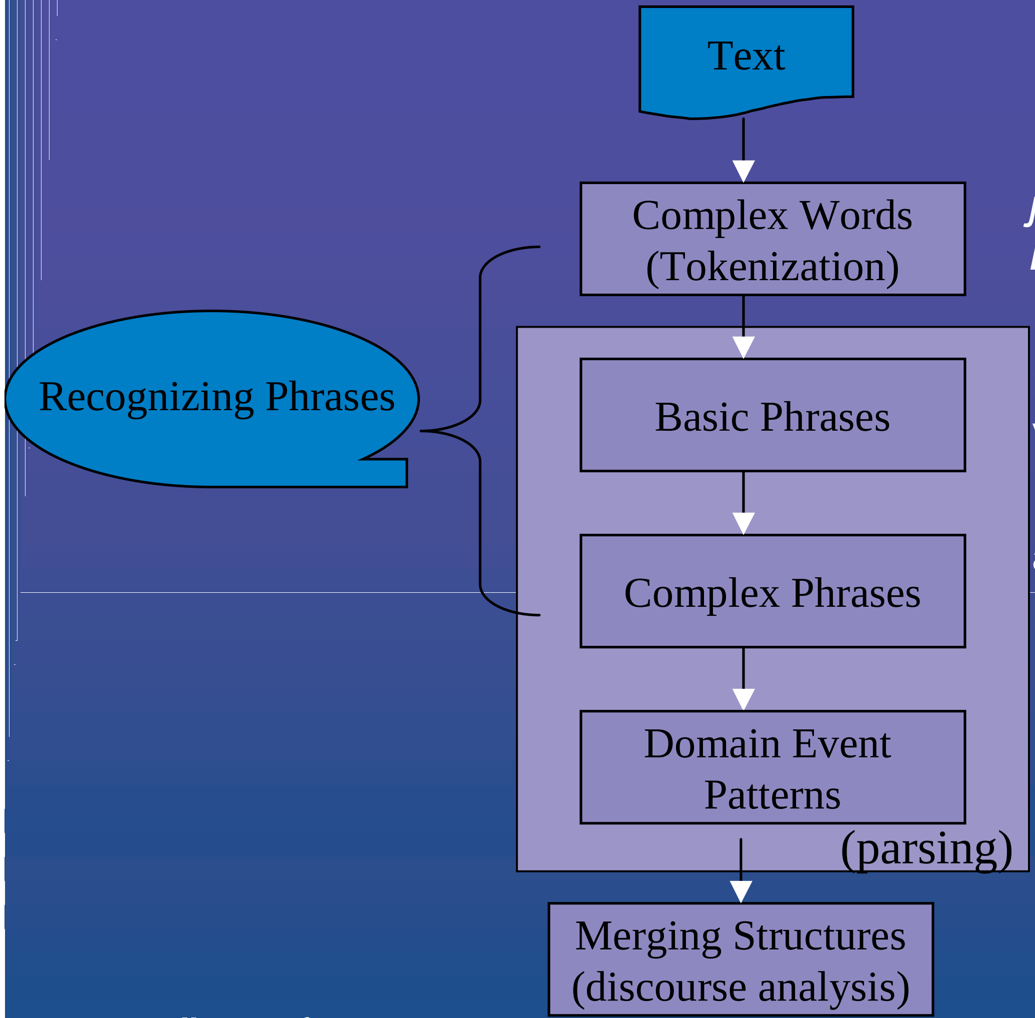


FASTUS

FASTUS

- Acronym for *Finite State Automaton Text Understanding System*
- Developed in SRI International, Menlo Park, California
- Joined MUC-4 (92), MUC-5 (93), MUC-6 (95), MUC-7 (97)
- English and Japanese
- Inspired by
  - the performance in MUC-3 that the group at the University of Massachusetts got out of a simple system [Lehnert et al., 1991]
  - Pereira's work on finite-state approximations of grammars [Pereira, 1990]
- Works as a set of cascaded and nondeterministic finite-state automata

FASTUS Architecture



joint venture, set up,
Bridgstone Sports Taiwan Co

noun groups: *a local concern*
verb groups: *had set up*

appositive attachment: *Peter, the director*
pp attachment: *production of iron and wood*

Example of Phrase Recognition

Salvadoran President-elect Afredo Cristiani condemned the terrorist killing of Attorney General Roberto Garcia Alvarado and accused the Farabundo Marti National Liberation Front (FMLN) of the crime

Noun Group:	**Salvadoran President-elect**
Name:	**Alfredo Cristiani**
Verb Group:	**condemned**
Noun Group:	**the terrorist killing**
Preposition:	**of**
Noun Group:	**Attorney General**
Name:	**Roberto Gareia Alvarado**
Conjunction:	**and**
Verb Group:	**accused**
Noun Group:	**the Farabundo Marti National Liberation Front (FMLN)**
Preposition:	**of**
Noun Group:	**the crime**

Example of Domain Event Pattern Recognition

[Salvadoran President-elect Afredo Cristiani][2] condemned [the terrorist killing of Attorney General Roberto Garcia Alvarado][1] and [accused the Farabundo Marti National Liberation Front (FMLN) of crime][2]

Two patterns are recognized:

1. **<Perpetrator> killing of <HumanTarget>**
2. **<GovtOfficial> accused <PerpOrg> of <Incident>**

Two incident structures are constructed:

Incident:	**KILLING**
Perpetrator:	**„terrorist“**
Confidence:	**__**
Human Target:	**„Roberto Garcia Alvarado“**

Incident:	**KILLING**
Perpetrator:	**FMLN**
Confidence:	**Accused by Authorities**
Human Target:	**__**

"pseudo-syntax"

- The material between the end of the subject noun group and the beginning of the main verb group must be read over, for example,

  ✓ Read over prepositional phrases and relative clauses

  1. Subject {Preposition NounGroup}* VerbGroup
  2. Subject Relpro {NounGroup | Other }* VerbGroup

  The mayor, who was kidnapped yesterday, was found dead today.

- Conjoined verb phrase, skipping over the first conjunct and associate the subject with the verb group in the second conjunct

Subject VerbGroup {NounGroup|Other}* Conjunction VerbGroup

Problem of "pseudo-syntax"

- **Same semantic content can be realized in different forms.**

  GM manufactures cars.

  Cars are manufactured by GM.
  ... GM, which manufactures cars ...
  ... cars, which are manufactured by GM ...
  ... cars manufactured by GM ...
  GM is to manufacture cars.
  Cars are to be manufactured by GM.
  GM is a car manufacturer.

- **Question: How many rules are needed to extract all relevant patterns? Why not using a linguistic theory?**
  - **Performance vs. Compotence**
  - **TACITUS: 36 hours to process 100 Messages**
  - **FASTUS: 12 minutes to process 100 messages**

Example of Template Merging

Salvadoran President-elect Afredo Cristiani condemned the terrorist killing of Attorney General Roberto Garcia Alvarado and accused the Farabundo Marti National Liberation Front (FMLN) of crime

Two incident structures are merged as

Incident:	KILLING
Perpetrator:	„terrorist“
Confidence:	—
Human Target:	„Roberto Garcia Alvarado

[redacted]

Incident:	KILLING
Perpetrator:	FMLN
Confidence:	Accused by Authorities
Human Target:	„Roberto Garcia Alvarado

GATE

Course: Intelligent Information Extraction

Neumann & Xu

GATE — a General Architecture for Text Engineering

- **An architecture**
A macro-level organisational picture for LE software systems.

- **A framework**
For programmers, GATE is an object-oriented class library that implements the architecture.

- **A development environment**
For language engineers, computational linguists et al, GATE is a graphical development environment bundled with a set of tools for doing e.g. Information Extraction.

- **Some free components...** ...and wrappers for other people's components

- **Tools** for: evaluation; visualise/edit; persistence; IR; IE; dialogue; ontologies; etc.

- **Free** software (LGPL). Download at http://gate.ac.uk/download/

DFKI IE Core Technologies

Course: Intelligent Information Extraction

Neumann & Xu

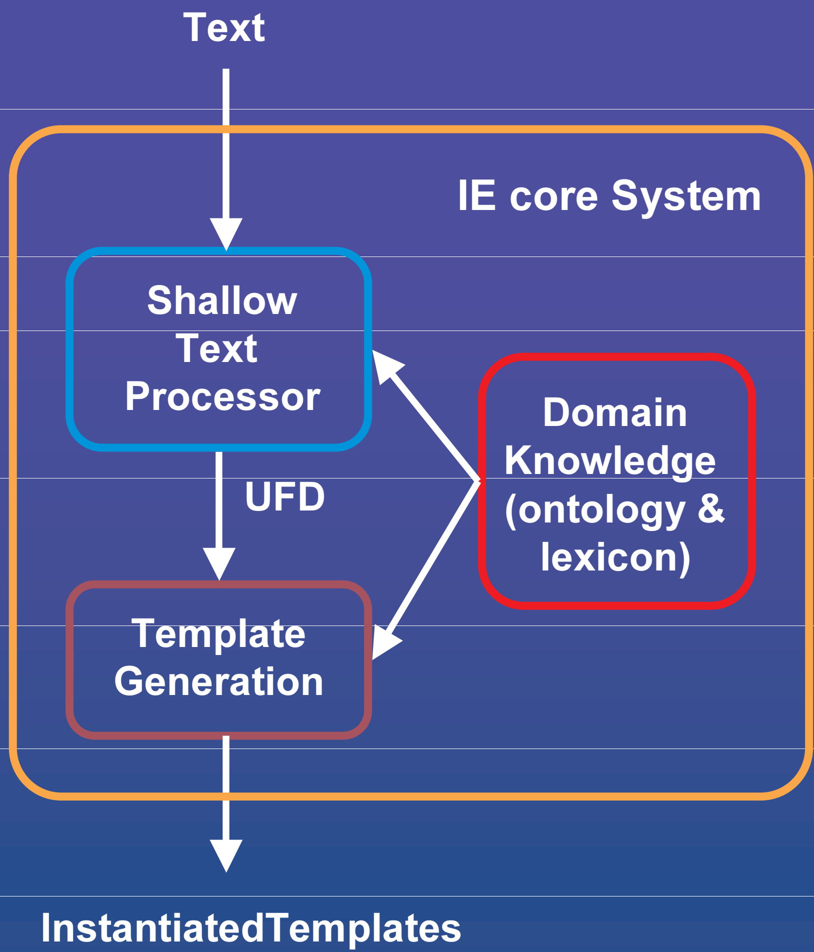
Information Extraction from real-world German text

SMES

Domain Adaptive IE Architecture

MAIN GOALS

- systematic treatment of domain-independent and domain-specific knowledge
  - robust and fast shallow text processor (mildly deep ☺)
  - abstract level of linking between linguistic and domain knowledge

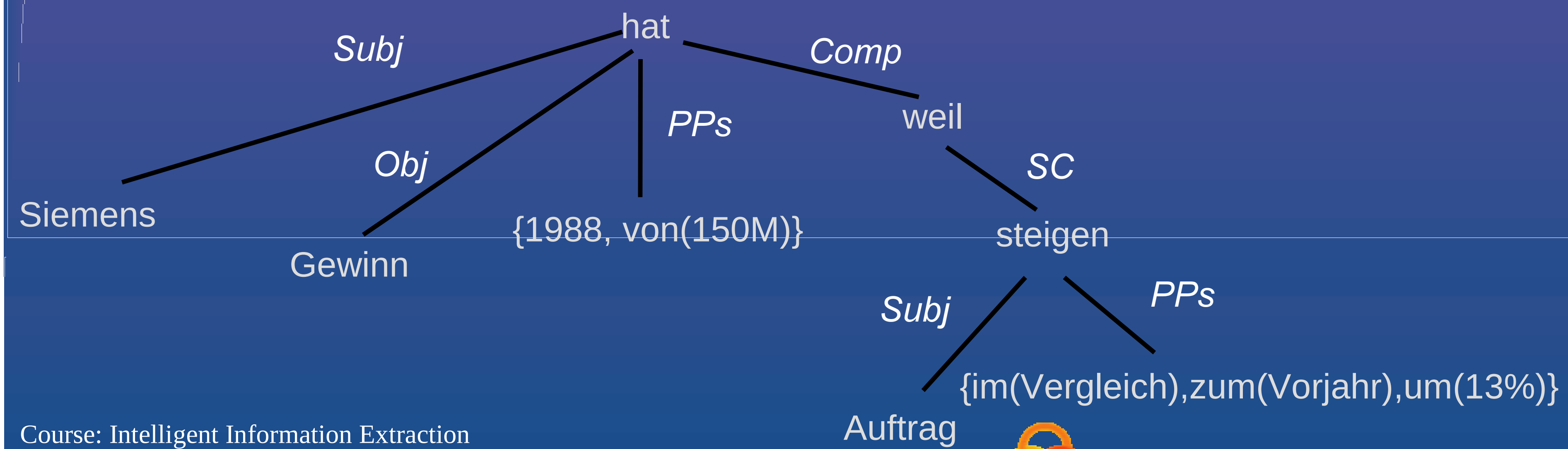


underspecified (partial) functional descriptions UFDs

UFD:
flat dependency-based structure, only upper bounds for attachment and scoping

[$_{\text{PN}}$Die Siemens GmbH] [$_{\text{V}}$hat] [$_{\text{year}}$1988][$_{\text{NP}}$einen Gewinn] [$_{\text{PP}}$von 150 Millionen DM], [$_{\text{Comp}}$weil] [$_{\text{NP}}$die Auftraege] [$_{\text{PP}}$im Vergleich] [$_{\text{PP}}$zum Vorjahr] [$_{\text{Card}}$um 13%] [$_{\text{V}}$gestiegen sind].

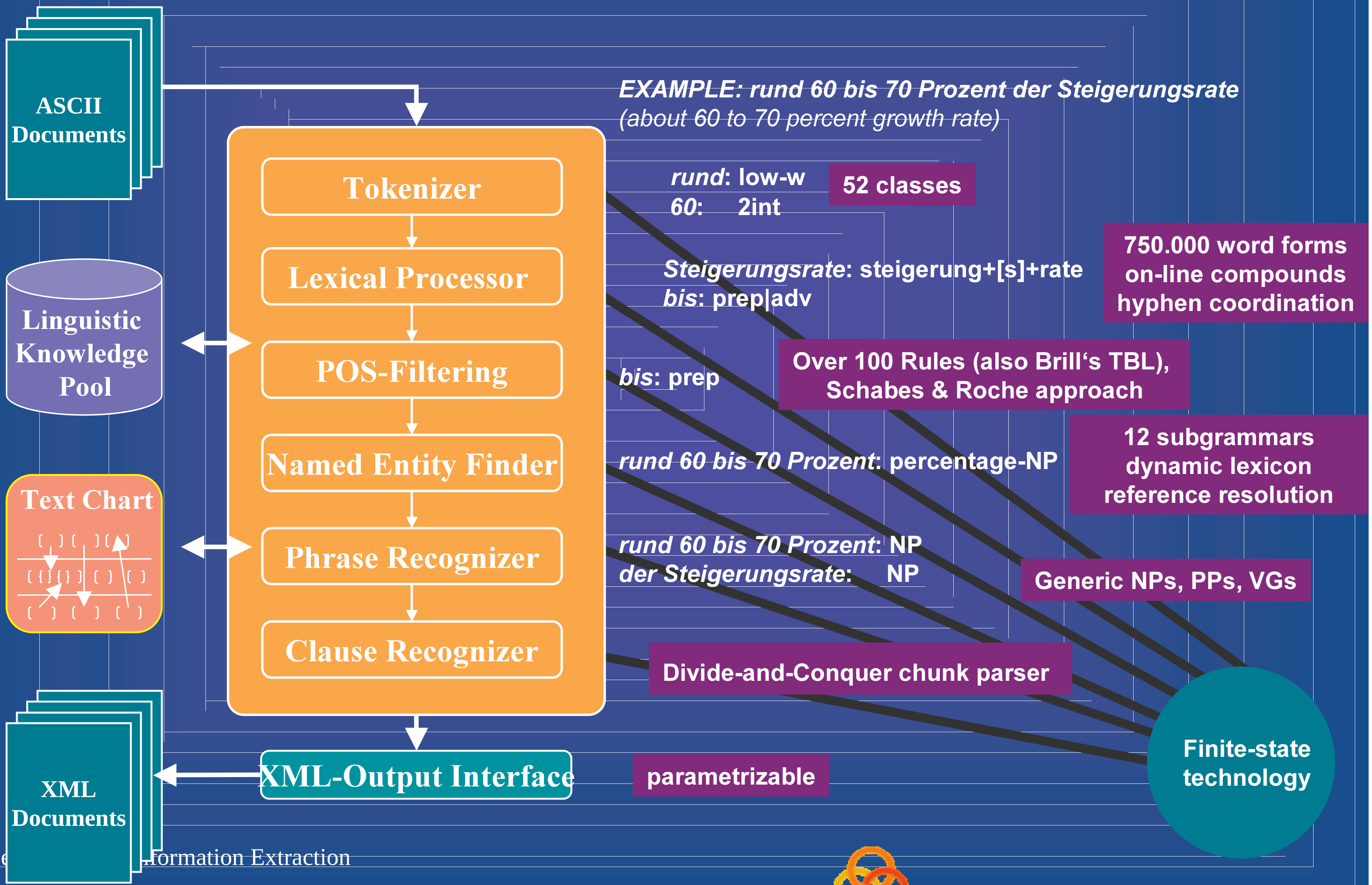
"The siemens company has made a revenue of 150 million marks in 1988, since the orders increased by 13% compared to last year."



Major Shallow Components

- C++ Toolkit for <u>weighted</u> finite state <u>transducers</u> (WFST)
  - e.g., determination, minimization, concatenation, local extension (following advanced approaches developed at AT&T, Xerox)
  - compact and efficient internal representations
- Dynamic tries for lexical & morphological processing
  - recursive traversal (e.g., for compound & derivation analysis)
  - robust retrieval (e.g., shortest/longest suffix/prefix)
- Parameterizable XML-output interface
- Both tools are portable across different platforms (Unix & Linux & Windows NT)

domain-independent shallow text processing components



Tokenizer

- The goal of the TOKENIZER is to:
  - map sequences of consecutive characters into word-like units (**tokens**)
  - identify the type of each token disregarding the context
  - performing word segmentation when necessary (e.g., splitting contractions into multiple tokens if necessary)
- overall more than 50 classes (proved to simplify processing on higher stages)
  - NUMBER_WORD_COMPOUND (“*69er*”)
  - ABBREVIATION (CANDIDATE_FOR_ABBREVIATION),
  - COMPLEX_COMPOUND_FIRST_CAPITAL („*AT&T-Chief*“)
  - COMPLEX_COMPOUND_FIRST_LOWER_DASH („*d'Italia-Chefs-“)*
- represented as single WFSA (406 KB)

Lexical Processor

- Tasks of the LEXICAL PROCESSOR:
  - retrieval of lexical information
  - recognition of **compounds** („Autoradiozubehör“ - *car-radio equipment*)
  - **hyphen coordination** („Leder-, Glas-, Holz- und Kunstoffbranche“ *leather, glass, wooden and synthetic materials industry)*
- lexicon contains currently more than 700 000 German full-form words (tries)
- each reading represented as triple <STEM,INFLECTION,POS>

  example: „wagen“ (*to dare* vs. *a car*)

STEM: „wag“
INFL: (GENDER: m,CASE: nom, NUMBER: sg)
(GENDER: m,CASE: akk, NUMBER: sg)
(GENDER: m,CASE: dat, NUMBER: sg)
(GENDER: m,CASE: nom, NUMBER: pl)
(GENDER: m,CASE: akk, NUMBER: pl)
(GENDER: m,CASE: dat, NUMBER: pl)
(GENDER: m,CASE: gen, NUMBER: pl)
POS: noun

STEM: „wag“
INFL: (FORM: infin)
(TENSE: pres, PERSON: anrede, NUMBER: sg)
(TENSE: pres, PERSON: anrede, NUMBER: pl)
(TENSE: pres, PERSON: 1, NUMBER: pl)
(TENSE: pres, PERSON: 3, NUMBER: pl)
(TENSE: subjunct-1, PERSON: anrede, NUMBER: sg)
(TENSE: subjunct-1, PERSON: anrede, NUMBER: pl)
(TENSE: subjunct-1, PERSON: 1, NUMBER: pl)
(TENSE: subjunct-1, PERSON: 3, NUMBER: pl)
(FORM: imp, PERSON: anrede)
POS: verb

Part-of-Speech Filtering

- The task of POS FILTER is to filter out unplausible readings of ambiguous word forms
- large amount of German word forms are ambiguous (20% in test corpus)
- contextual filtering rules (ca. 100)
  - example:

„Sie <u>bekannten</u>, die <u>bekannten</u> Bilder gestohlen zu haben“

They confessed they have stolen the famous pictures

„bekannten“ - *to confess* vs. *famous*

FILTERING RULE: if the <u>previous word form is determiner</u> and the <u>next word form is a noun</u> then <u>filter out the verb reading</u> of the current word form

- supplementary rules determined by Brill‘s tagger in order to achieve broader coverage
- rules represented as FSTs, hard-coded rules (filtering out rare readings)

Named Entity Finder

- The task of the NAMED ENTITY FINDER is the identification of:
  - entities: organizations, persons, locations
  - temporal expressions: time, date
  - quantities: monetary values, percentages, numbers
- Identification in two steps:
  - recognition patterns expressed as WFSA are used to identify phrases containing potential candidates for named entities
  - additional constraints (depending on the type of a candidate) are used for validating the candidates and an appropriate extraction rule is applied in order to recover the named entity

  example: „von knapp neun Milliarden auf über 43 Milliarden Spanische Pesetas“
  from almost nine billions to more than 43 billions spanish pesetas

  TYPE: monetary
  SUBTYPE: monetary-prepositional-phrase

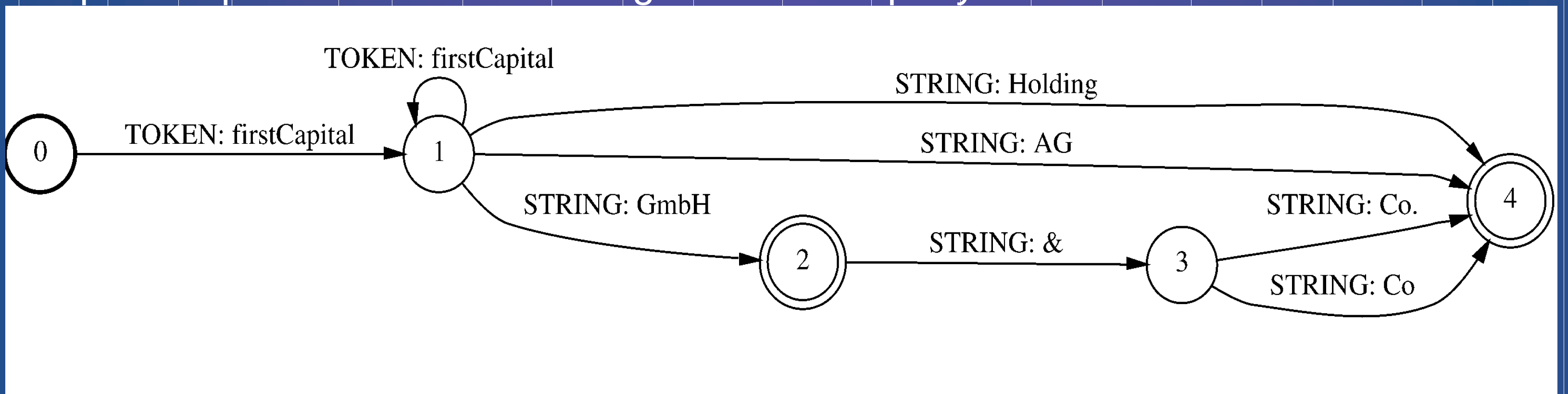
- Longest match strategy

Named Entity Finder (cont.)

- Arcs of the WFSAs are predicates on lexical items:

  (a) **STRING: s**, holds if the surface string mapped by current lexical item is of the form **s**
  (b) **STEM: s**, holds if: the current lexical item has a preferred reading with stem **s** or the current lexical item does not have preferred reading, but at least one reading with stem **s**
  (c) **TOKEN: x**, holds if the token type of the surface string mapped by current lexical item is **x**

- Example: simple automaton for recognition of company names



additional constraint: disallow determiner reading for the first word
candidate: „Die Braun GmbH & Co.“ extracted: „Braun GmbH & Co.“

Named Entity Finder (cont.)

- Additional lexica for geographical names, first names (persons) and company names compiled as WFSA (new token classes)
- Named entities may appear without designators (companies, persons)
- Dynamic lexicon for storing named entities without designators
- Candidates for named entities, example:

*Da flüchten sich die einen ins Ausland, wie etwa der Münchner Strickwarenhersteller **März GmbH** oder der badische Strumpffabrikant Arlington Socks, GmbH. Ab kommendem Jahr strickt **März** knapp drei Viertel seiner Produktion in Ungarn.*

- Resolution of type ambiguity using the dynamic lexicon:
  if an expression can be a person name or company name (*Martin Marietta Corp.*)
  then use type of last entry inserted into dynamic lexicon for making decision

Performance of Shallow Text Processing for German

- **Basis**

  corpus of German business magazine „Wirtschaftswoche“ (1,2MB, 197118 tokens)

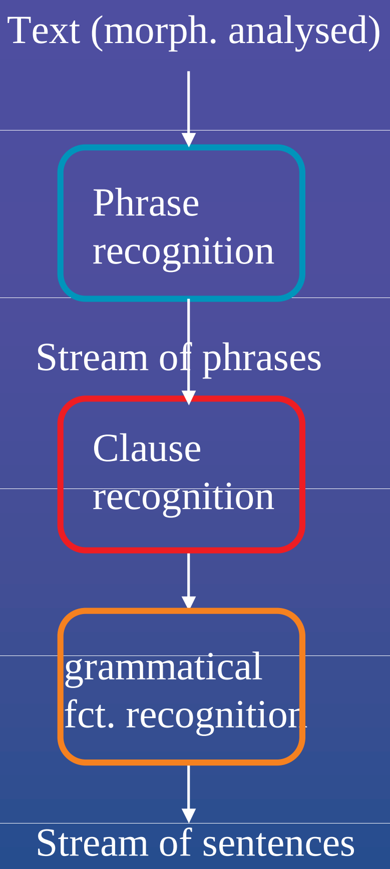
- **Performance**

  ~32sec. (~6160 wrds/sec; PentiumIII, 500MHz, 128Ram)

- **Evaluation** (20.000 tokens)

	Recall	Precision
• compound analysis:	98.53%	99.29%
• part-of-speech-filterung:	74.50%	96.36%
• Named entity (including NE reference resolution; all 85% R, 95.77% P)		
• person names:	81.27%	95.92%
• companies:	67.34%	96.69%
• locations:	75.11%	88.20%
• total:	73.94%	94.10%
• fragments (NPs, PPs):	76.11%	91.94%

chunk parsing strategies for the improvment of robustness and coverage on the sentence level



Current chunk parser

bottom-up:
first phrases and then sentence structure

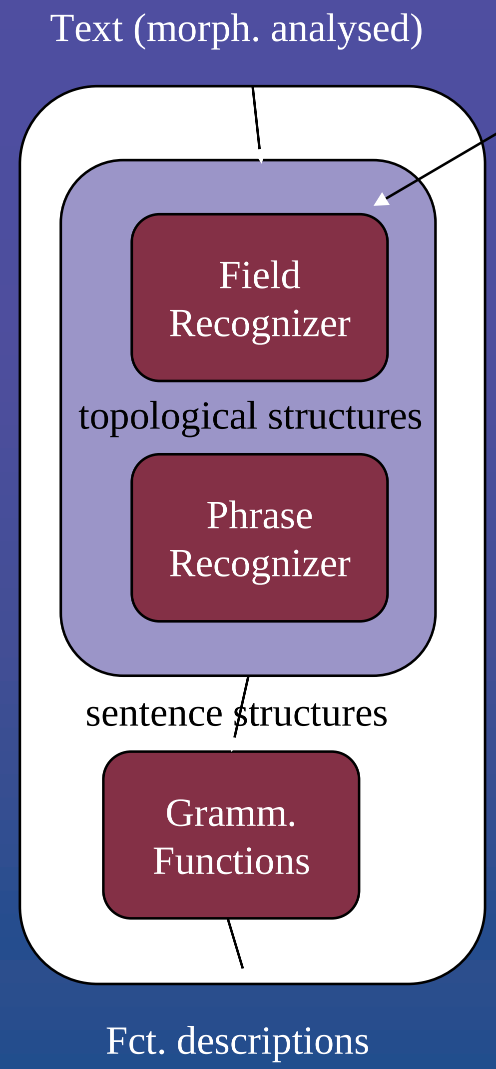
main problem:
even recognition of simple sentence structure depends on performance of phrase recognition

example:
- complex NP (*nominalization style*)
- relative pronouns

[Die vom Bundesgerichtshof und den Wettbewerbern als Verstoss gegen das Kartellverbot gegeisselte zentrale TV-Vermarktung] ist gängige Praxis.
(*[central television marketing censured by the German Federal High Court and the guards against unfair competition as an act of contempt against the cartel ban] is common practice*)

A new chunk parser

Divide-and-conquer strategy



first compute topological structure of sentence
second apply phrase recognition to the fields

[$_{coord}$ [$_{core}$ *Diese Angaben konnte der Bundesgrenzschutz aber nicht bestätigen*], [$_{core}$ *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.)

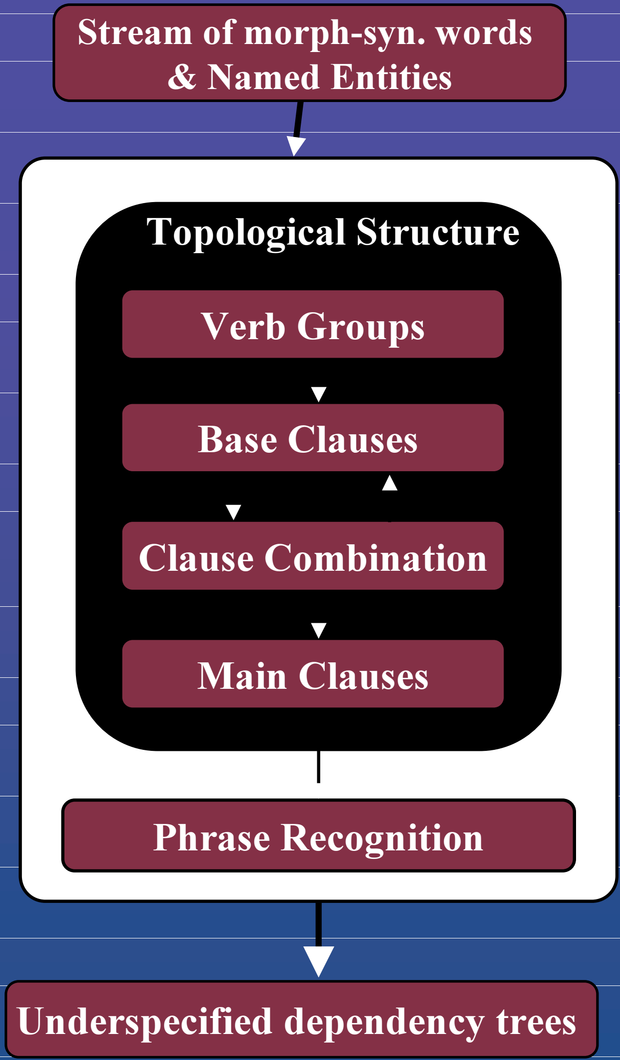
Evaluation

400 sentences (6306 words)

Verb groups:	98.59% F
Clause struct.:	91.62% F
All components:	87.14% F

(more details in Master thesis of C. Braun and NeumannBraunPiskorski:ANLP00)

The divide-and-conquer parser: a series 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.

Weil die Siemens GmbH, Rel-Clause Verluste Verb-FIN, Modv-FIN sie Aktien FV-Inf.

Subconj-Clause, Modv-FIN sie Aktien FV-Inf.

Clause

Advanced Multilingual Information Extraction with SProUT

–

Shallow Processing with Unification and Typed Feature Structures

Motivation for SProUT

- one platform for development of multilingual STP systems
  - development & testing environment for resources
  - well-documented programming API
- flexible interface to different processing modules
  - TFSs as abstract interchange format
  - XML-based representation
- find good trade-off between efficiency and expressiveness
  - word-based finite-state devices
  - typed unification-based grammars

SProUT - XTDL Formalism

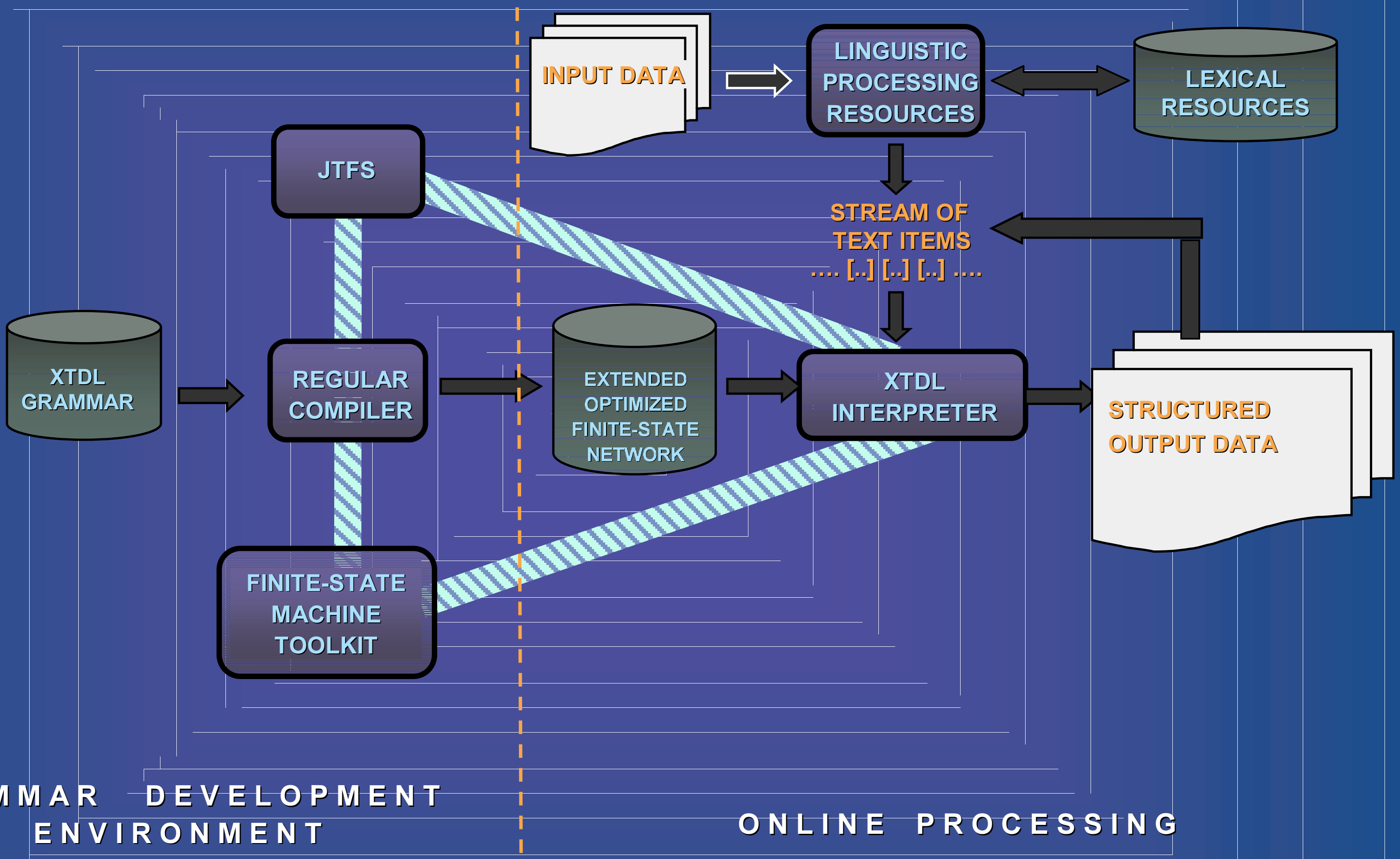
→ Combines typed feature structures (TFS) and regular expressions, including
coreferences and functional application

→ XTDL grammar rules – production part on LHS, and output description on RHS

→ Couple of standard regular operators:

concatenation		optionality	?
disjunction	|	Kleene star	*
Kleene plus	+	n-fold repetition	{n}
m-n span repetition	{m,n}	negation	~

SProUT Architecture



SProUT - XTDL Formalism

```
pp :> morph & [POS Prep, SURFACE #prep, INFL [CASE #c]]
      morph & [POS Det, INFL [CASE #c, NUMBER #n, GENDER #g]] ?
      morph & [POS Adjective,INFL [CASE #c, NUMBER #n, GENDER #g]] *
      morph & [POS noun, SURFACE #noun_1,
                        INFL [CASE #c, NUMBER #n, GENDER #g]]
      morph & [POS noun, SURFACE #noun_2,
                        INFL [CASE #c, NUMBER #n, GENDER #g]] ?
  -> phrase & [CAT pp,
                        PREP #prep
                        AGR agr & [CASE #c, NUMBER #n, GENDER #g],
                        CORE_NP #core_np],
      where #core_np = Append(#noun_1," ",#noun_2).
```

SProUT - XTDL Grammar Processing

- Three-step processing
  - Matching of regular patterns using unifiability (LHS)
  - Rule instance creation
  - Unification of the rule instance and matched input stream
- Longest match strategy
- Ambiguities allowed
- Optimization techniques
  - Problem: TFSs treated as symbolic values by FSM Toolkit
  - Sorting outgoing transitions from selected states (transition hierarchy under subsumption)
- Rule prioritization
- Output merging

SProUT - Processing Resources

- Tokenization
  - Fine-grained token classification (over 30 classes)
  - Domain and language specific token subclassification

$$\begin{bmatrix} \text{SURFACE}: \text{"Berlin"} \\ \text{TYPE}: \text{first\_capital\_word} \\ \text{SUBTYPE}: \text{city\_sufix} \end{bmatrix}_{token}$$

  - Token postsegmentation
  - Supports almost all European character sets
  - Minor adjustments necessary while porting to new language

e.g. word_with_apostrophe class: *it's -> it* + ' + s

SProUT - Processing Resources

- Morphology
  - Full-form lexica (Mmorph) vs. external morphology components
  - On-line shallow compound recognition
  - Current coverage

Language	Coverage	Extras
English	200,000	
German	830,000	compound recognition
French	225,000	
Dutch	370,000	compound recognition
Italian	330,000	
Spanish	570,000	
Polish	120,000 lexemes (ca. 2 Mil. word forms)	
Czech	600,000	Morph. Disambiguation
Japanesese	ca. 98% F-measure	Word segmentation + POS_tagger
Chinese		Word segmentation + POS_tagger

SProUT - Processing Resources

→ Gazetteer

- Allows for associating entries with a list of arbitrary attribute-value pairs

Argentyny | GAZ_TYPE: country | CONCEPT: Argentyna | FULL_NAME: Republika Argentyny
| CASE: genitive | CAPITAL: Buenos Aires | continent .....

$$\begin{bmatrix} \text{SURFACE}: \text{"Washington"} \\ \text{TYPE}: \text{gaz\_surname} \end{bmatrix}_{gazetteer}, \begin{bmatrix} \text{SURFACE}: \text{"Washington"} \\ \text{TYPE}: \text{gaz\_city} \\ \text{CONCEPT}: \text{c\_washington\_cit} \end{bmatrix} \begin{bmatrix} \text{SURFACE}: \text{"Washington"} \\ \text{TYPE}: \text{gaz\_state} \\ \text{CONCEPT}: \text{c\_washington\_state} \end{bmatrix}_{gazetteer}$$

- Reusage of available resources across languages

SProUT - Processing Resources

- Reference Matcher
  - Finds identity relation between recognized entities and unconsumed text
    fragments

    …. *CEO Prof. Dr. Martin Marietta mentioned new take-over*……….

    ……………………………………………………………………………..

    ……………*Martin Marietta GmbH plans to*……………………………..

    ………………………………………………………………………….

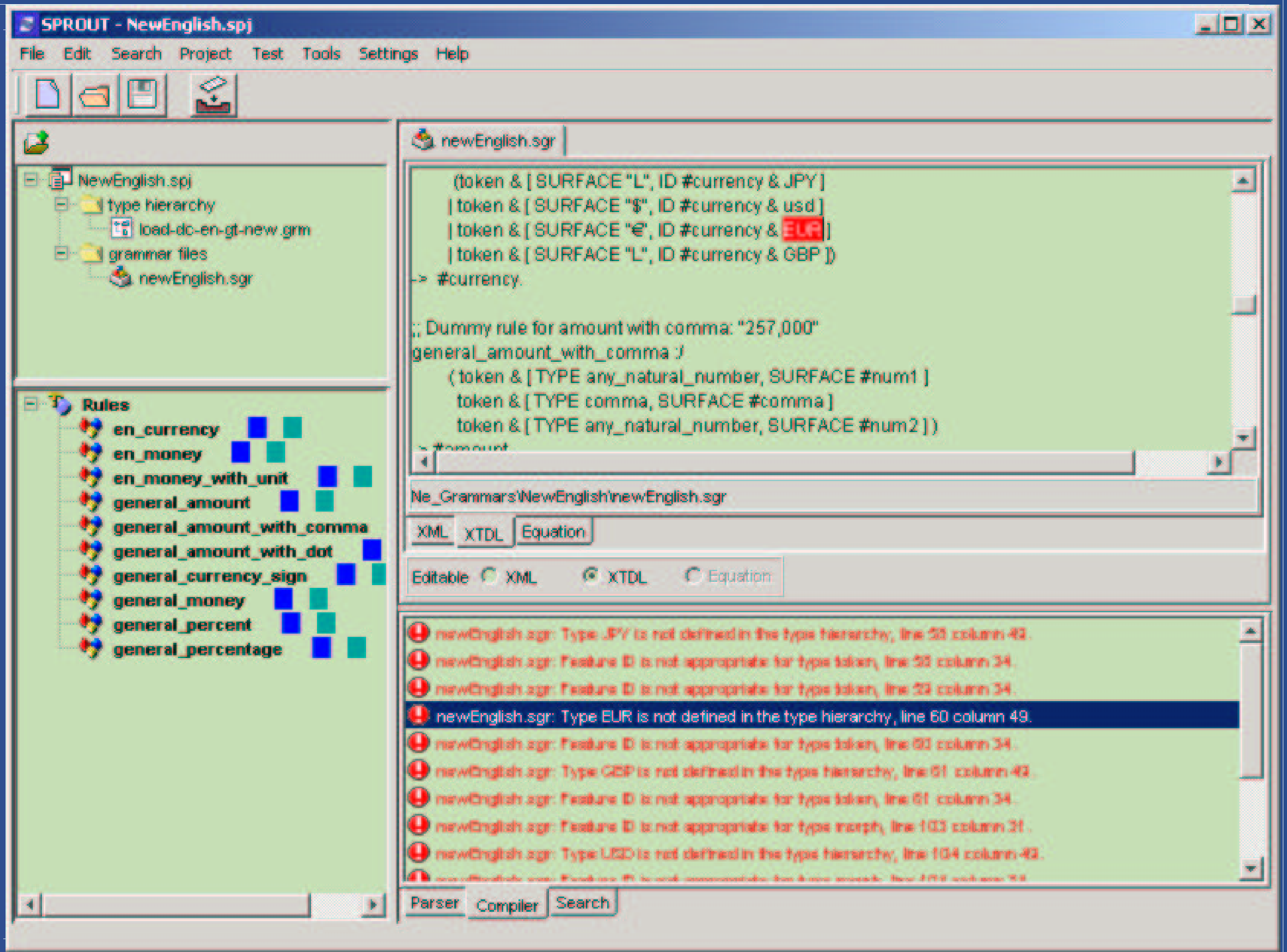
    ………………… *Marietta said* ……………………………………….....

  - Grammarians define how variants are built

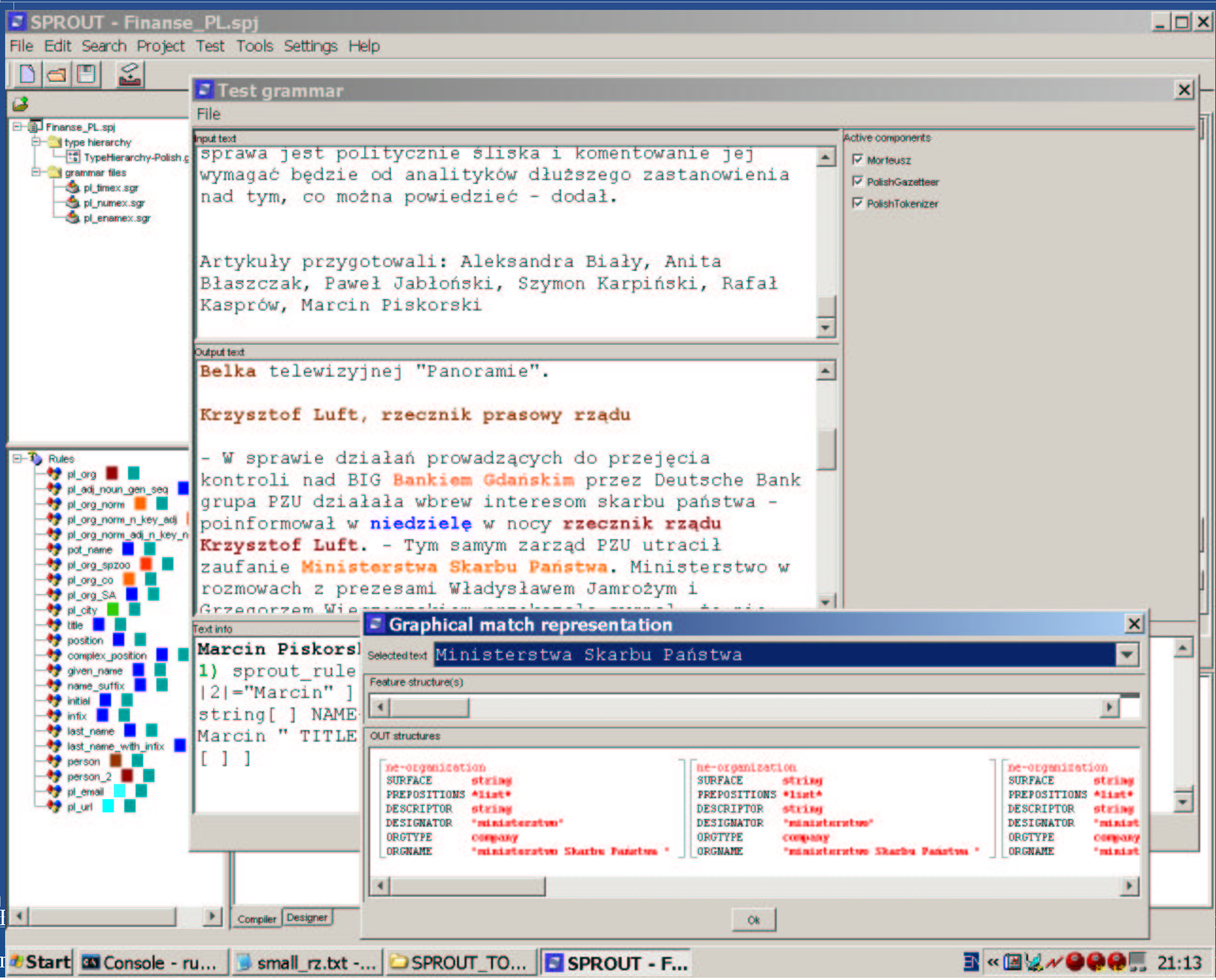
```
person :> ....... #first_name ...... #last_name ..... #title ...
       -> [FIRST_NAME #first_name, LAST_NAME #last_name,
           TITLE #title,VARIANT #variant],
        where #variant = { Conc(#title,#last_name), #last_name }
```

  - Parametrizable Contextual Frame

SProUT - IDE

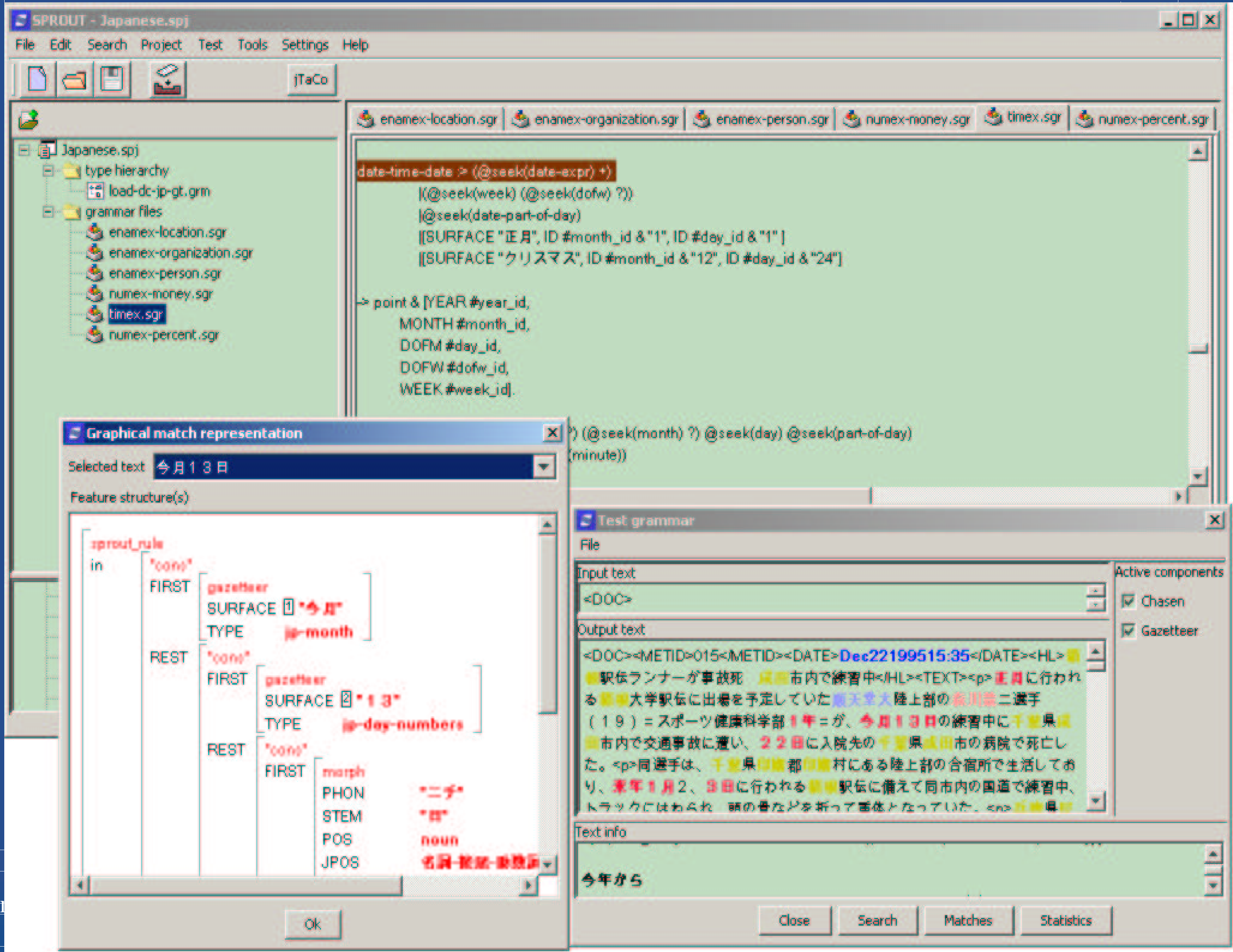

SPROUT - NewEnglish.spj
File Edit Search Project Test Tools Settings Help
NewEnglish.spj
type hierarchy
load-dc-en-gt-new.grm
grammar files
newEnglish.sgr
Rules
en_currency
en_money
en_money_with_unit
general_amount
general_amount_with_comma
general_amount_with_dot
general_currency_sign
general_money
general_percent
general_percentage
newEnglish.sgr
(token & [SURFACE "L", ID #currency & JPY]
| token & [SURFACE "$", ID #currency & usd]
| token & [SURFACE "€", ID #currency & EUR]
| token & [SURFACE "L", ID #currency & GBP])
-> #currency.
;; Dummy rule for amount with comma: "257,000"
general_amount_with_comma :/
(token & [TYPE any_natural_number, SURFACE #num1]
token & [TYPE comma, SURFACE #comma]
token & [TYPE any_natural_number, SURFACE #num2])
Ne_Grammars\NewEnglish\newEnglish.sgr
XML XTDL Equation
Editable XML XTDL Equation
newEnglish.sgr: Type EUR is not defined in the type hierarchy, line 60 column 49.
Parser Compiler Search

SProUT - IDE



SPROUT - Finanse_PL.spj
File Edit Search Project Test Tools Settings Help
Test grammar
File
Input text
sprawa jest politycznie śliska i komentowanie jej
wymagać będzie od analityków dłuższego zastanowienia
nad tym, co można powiedzieć - dodał.
Artykuły przygotowali: Aleksandra Biały, Anita
Błaszczak, Paweł Jabłoński, Szymon Karpiński, Rafał
Kasprów, Marcin Piskorski
Active components
Morfeusz
PolishGazetteer
PolishTokenizer
Output text
Belka telewizyjnej "Panoramie".
Krzysztof Luft, rzecznik prasowy rządu
- W sprawie działań prowadzących do przejęcia
kontroli nad BIG Bankiem Gdańskim przez Deutsche Bank
grupa PZU działała wbrew interesom skarbu państwa -
poinformował w niedzielę w nocy rzecznik rządu
Krzysztof Luft. - Tym samym zarząd PZU utracił
zaufanie Ministerstwa Skarbu Państwa. Ministerstwo w
rozmowach z prezesami Władysławem Jamrożym i
Finanse_PL.spj
type hierarchy
grammar files
pl_timex.sgr
pl_numex.sgr
pl_enamex.sgr
Rules
pl_org
pl_adj_noun_gen_seq
pl_org_norm
pot_name
pl_org_spzoo
pl_org_co
pl_org_SA
pl_city
title
position
complex_position
given_name
name_suffix
initial
infix
last_name
last_name_with_infix
person
person_2
pl_email
pl_url
Text info
Marcin Piskors
1) sprout_rule
|2|="Marcin"]
string[ ] NAME
Marcin " TITLE
[ ]]
Compiler
Designer
Graphical match representation
Selected text
Ministerstwa Skarbu Państwa
Feature structure(s)
OUT structures
ne-organization
SURFACE string
PREPOSITIONS *list*
DESCRIPTOR string
DESIGNATOR "ministerstwo"
ORGTYPE company
ORGNAME "ministerstwo Skarbu Państwa "
Ok
Start
Console - ru...
small_rz.txt -...
SPROUT_TO...
SPROUT - F...
21:13

SProUT Development Environment



SProUT – Implementation & Availability

- Implementation
  - Core Components:
    - FSM Toolkit: **C++ (JAVA APIs)**
    - JTFS: **JAVA**
    - XML-based General Purpose Regular Compiler: **JAVA, C++**
  - Linguistic Processing Components: **JAVA**
  - Integrated Development Environment : **JAVA**
- Availability
  - IDE + Core Components: **Windows, Linux**
  - **JAVA APIs** for integration in other frameworks
  - Freeware **?**

SProUT – Applications

- Industrial projects
  - CCA – Customer Care Automation (Telekom)

    Q/A System with dialog facilities for telecommunication domain
  - AGRO Server (Celi S.R.L.)

    Monitoring and Mining Customer Orientation

    http://www.celi.it

- Internal projects
  - Extralink

    Information System combining Information Extraction and Hyperlinking for Touristic domain

SProUT – Applications

- EU-funded projects
  - AIRFORCE (AIR FOReCast in Europe)
    - Tools for for forecasting air traffic in Europe
  - MEMPHIS (Multilingual Content for Flexible Format Internet Premium Services)
    - Platform for cross-lingual premium content services for thin clients
    - http://www.ist-memphis.org/
  - DEEP-THOUGHT
    - Hybrid system combining shallow and deep methods for knowledge intensive information extraction
    - http://www.eurice.de/deepthought/

Combining Deep and Shallow NLP for IE

Course: Intelligent Information Extraction

Neumann & Xu

Isn‘t SNLP enough?

- Many simple NL tasks only need SNLP
  - E.g., text clustering, Named Entity recognition, term extraction, binary relation extraction
- Information extraction: 60% f-measure barrier in case of scenario template extraction (e.g., *management succession).* Complex phenomena:
  - Grammatical function/deep case recognition
  - Free word order in German
  - Multiple sentences
  - Reference resolution for template merging
- Content-oriented Web-services/Semantic Web
  - More precise structural extraction needed
  - E.g., ontology extraction, question/answering systems
- Shallow systems are getting deeper, requesting more and more accurate linguistic information
  - integration of deep NL components

Possible Integration Scenario

The parallel approach

- Run SNLP and DNLP in parallel
  - prefer results from DNLP
- Problem: different processing speed when processing large quantities of text:
  - Precision: slowest component ⇒ effects overall speed
  - Speed: overall precision hardly distinguishable from shallow-only system ⇒ DNLP always time out

The Whiteboard approach

- Integrated, flexible architecture where components can play at their strengths
- Call DNLP on results of SNLP:
  - Results from SNLP used to identify relevant candidates for on demand use of DNLP (domain-specific criteria)
  - Robustness handled by SNLP to increase the coverage of DNLP
  - Use SNLP to guide DNLP towards the most likely syntactic analysis leading to improved speed-up

Combining the Best of the Two Worlds for IE

- utilization of SNLP as primary linguistic resources for
  - robustness, efficiency and domain-specific interpretation of local relationships
- integration of DNLP, if the analysis exists and it is relevant, for extracting
  - relationships embedded in complex linguistic constructions,e.g., free word order, long distance dependencies, control and raising, or passive
- combination of finite-state technology with unification-based formalism for
  - definition of template filling rules, scenario templates and template elements
  - using unification and subsumption check as basic operations for template merging

A Simple Example

After the retirement of Peter Smith,

Mary Hopp was persuaded to take over the development sector.

shallow: retirement of $\boxed{1}$ PN $\longrightarrow$ [Person_Out $\boxed{1}$]

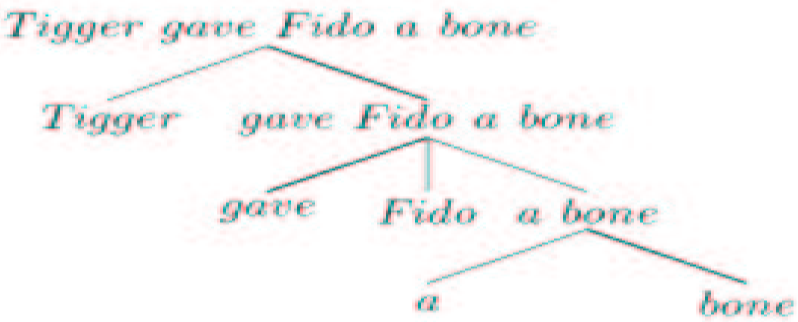
deep:

$$\begin{pmatrix} \text{Pred} & & \text{„take over“} \\ \text{Agent} & \boxed{2} & \text{„Mary Hopp“} \\ \text{Theme} & \boxed{3} & \text{„development sector“} \end{pmatrix} \longrightarrow$$

$$\begin{pmatrix} \text{Person\_In} & \boxed{2} \\ \text{Sector} & \boxed{3} \end{pmatrix}$$

Head-Driven Pharse Structure Grammar (HPSG)

- A constraint-based, lexicalist approach to grammatical theory
- Model languages as systems of constraints
- Linguistic units and their relations are expressed by typed feature structures
- No transformations for unbounded dependencies
- Multiple inheritance hierarchies for lexicon organization
- More information see
  - http://lingo.stanford.edu/erg.html
  - http://www.coli.uni-sb.de/%7Ehansu/courses/hpsg_arch.html
  - http://www.coli.uni-sb.de/%7Ehansu/psfiles/hpsg.ps



$$\begin{bmatrix} \text{PHON} & \langle \textit{Tigger gave Fido a bone} \rangle \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR} & \begin{bmatrix} \text{PHON} & \langle \textit{gave Fido a bone}\rangle \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR} & [\text{PHON } \langle \textit{gave}\rangle] \\ \text{COMP-DTRS} & \left\langle [\text{PHON } \langle \textit{Fido}\rangle], \begin{bmatrix} \text{PHON} & \langle \textit{a bone}\rangle \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR} & [\text{PHON } \langle \textit{bone}\rangle] \\ \text{COMP-DTRS} & \langle [\text{PHON } \langle \textit{a}\rangle] \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{COMP-DTRS} & \langle [\text{PHON } \langle \textit{Tigger}\rangle] \rangle \end{bmatrix} \end{bmatrix}$$

Architecture of HPSG

The Organisation of a Grammar

Grammar consists of

- Language universal principles
- Language specific principles
- Grammar rules (Language specific)
- Lexicon

Universal Grammar:

$$\mathrm{UG} = P_1 \wedge \ldots \wedge P_n$$

Language Specific Grammars:

$$\mathrm{English} = P_1 \wedge \ldots \wedge P_{n+1} \ldots \wedge P_{n+m} \wedge (L_1 \vee \ldots \vee L_p \vee R_1 \vee \ldots \vee R_q)$$

$$\mathrm{French} = P_1 \wedge \ldots \wedge P'_{n+1} \ldots \wedge P'_{n+m} \wedge (L'_1 \vee \ldots \vee L'_p \vee R'_1 \vee \ldots \vee R'_q)$$

Subject control verb *promise*

$$\begin{bmatrix} \text{PHON} & \langle \textit{promises} \rangle \\ \text{SYNSEM|LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{MAJ} & \textit{Verb} \\ \text{VFORM} & \textit{fin} \\ \text{AUX} & - \\ \text{INV} & - \end{bmatrix} \\ \text{LEX} & + \\ \text{SUBCAT} & \left\langle \mathrm{NP}_{\boxed{1}}, \mathrm{NP}_{\boxed{2}}, \mathrm{VP}\begin{bmatrix} \text{VFORM} & \textit{inf} \\ \text{SUBCAT} & \langle \mathrm{NP}_{\boxed{1}} \rangle \end{bmatrix}_{\boxed{3}} \right\rangle \end{bmatrix} \\ \text{CONTENT} & \begin{bmatrix} \text{RELN} & \textit{promise} \\ \text{AGENT} & \boxed{1} \\ \text{PATIENT} & \boxed{2} \\ \text{THEME} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

WHIE: A Hybrid IE Approach

- hybrid information extraction architecture built on top of Whiteboard Annotation Machine (WHAM)
- domain modeling for hybrid architecture
  - semi-automatic construction of template filling rules
  - heuristics for domain/relevance-driven access to different levels of WHAM
- evaluation
  - hybrid approach
  - improvement after integration of deep NLP

Integration of Shallow and Deep Analysis

Goal of integrated 'hybrid' syntactic processing

- *Robustness and efficiency* of shallow analysis
- *Precision and fine-grainedness* of deep syntactic analysis

Lexical Integration

- SPPC-HPSG interface: building HPSG lexicon entries "on the fly"
  - Named entities, open class categories (nouns, adjectives, adverbs, ..)
- HPSG-GermaNet integration (Siegel, Xu, Neumann 2001)
  - association with HPSG lexical sorts

➢ *coverage and robustness*

Phrasal integration for 'hybrid' syntactic processing

- Robust, stochastic topological field parsing
- Integration of shallow topological and deep syntactic parsing

➢ *efficiency and robustness*

Integration of Shallow and Deep Analysis

— Problems and Solutions —

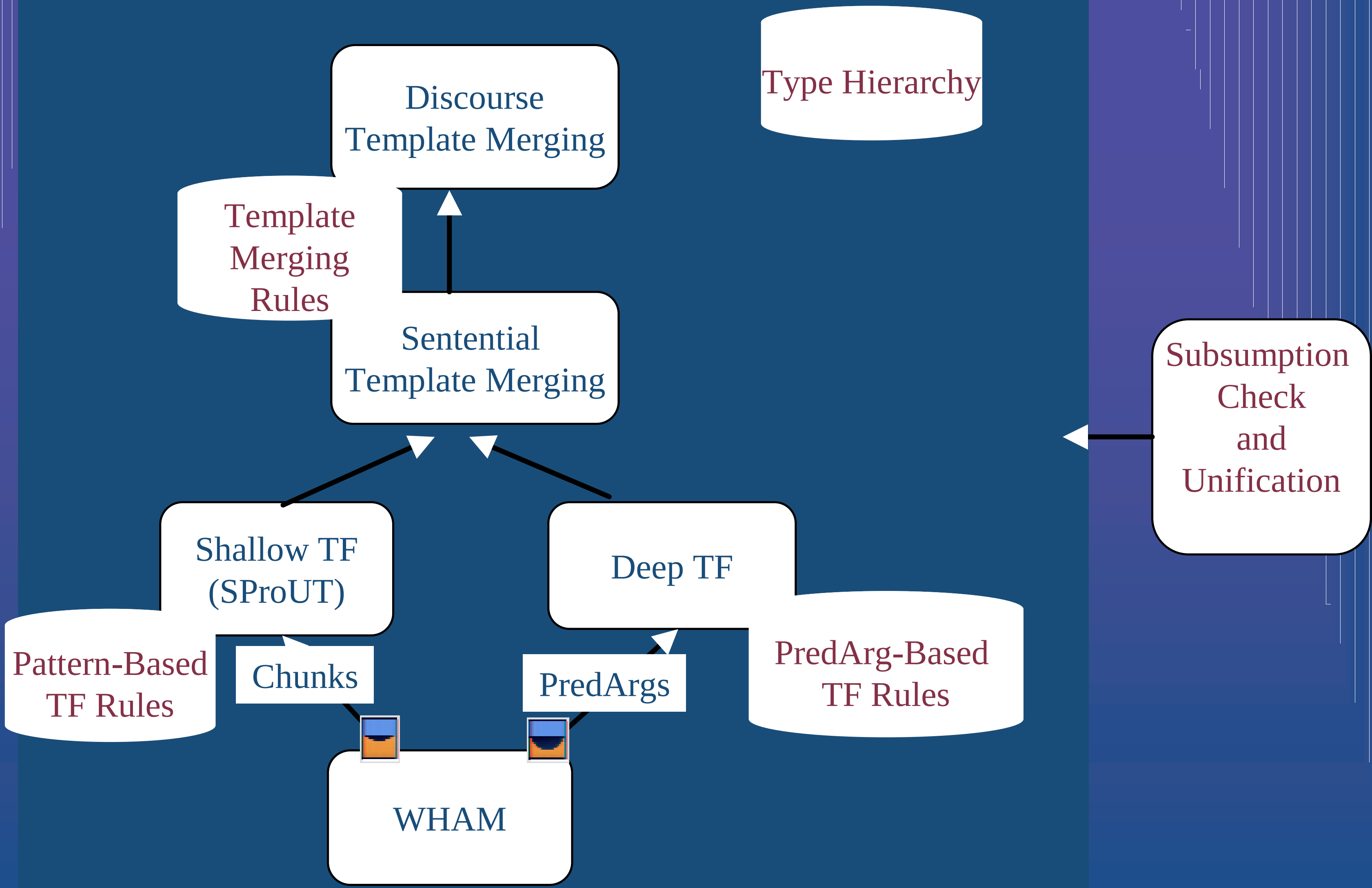
Integrated of shallow and deep parsing

- Guiding deep analysis by shallow (chunk-based) pre-partitioning
- Interleaved (hybrid) shallow/deep processing

The deep/shallow mapping problem

- Chunk parsing not isomorphic to deep syntactic structure („attachments“)

Template Extraction System
(Xu & Krieger, 2003)



Typed Feature Structure (TFS) for IE task

- **As a formal language for the modeling of the domain vocabulary and their relations in the typed hierarchy**
  - Template elements
  - Scenario templates
  - Linguistic units
  - Domain ontologies
  - Rules for template filling and template merging
- **Dynamic online construction of templates**
- **TFS unifier supports wellformed unification of TFSs and subsumption check**

Template filling with deep and shallow analysis (Example)

Interaction of passive, control and multiple named entities.

Because of the <u>retirement von Peter Müller,</u>
Hans Becker was persuaded to take over the development sector.

<u>Pattern based TF rule (Shallow TF)</u>

retirement•von• [1] PN → PO [1]

PERSON_OUT "Peter Müller"

<u>Deep TF Rule</u>

PERSON_IN [1]
DIVISION [2]
DEEP
PRED "take over"
AGENT [1] "Hans Becker"
THEME [2] "development sector "

PERSON_OUT "Peter Müller"
PERSON_IN "Hans Becker"
DIVISION "development sector"
ORGANISATION
POSITION

Semi-Automatic Construction of Template Filling Rules

- learning relevant terms (Xu et al. 2002)
  - a KFIDF term-classification method
  - verb-noun collocations

- adaptation of relevant terms to hybrid architecture
  - **Shallow**: adjectives and nouns as triggers for pattern-based template filling rules, e.g.,

    The previous president of car supplier Kolbenschmidt, Heinrich Binder

    Successor of the officeholder Hans Guenter Merk
  - **Deep**: verbs as triggers for lexicalized unification-based template filling rules, e.g.,

    General manager Eugen Krammer (59), …, will resign from his office on May 31. 1997

⇨ if a domain is dominated by relevant verbs, it is helpful to integrate DNLP

⇨ domains can be characterized by the distribution of terms with respect to POSs

Single-Word Term Extraction

a specific TFIDF measure: KFIDF

a word is relevant if it occurs more frequently than other words in a certain category and rarely in other categories.

$$KFIDF(w, cat) = docs(w, cat) \times LOG\left(\frac{n \times |cats|}{cats(w)} + 1\right)$$

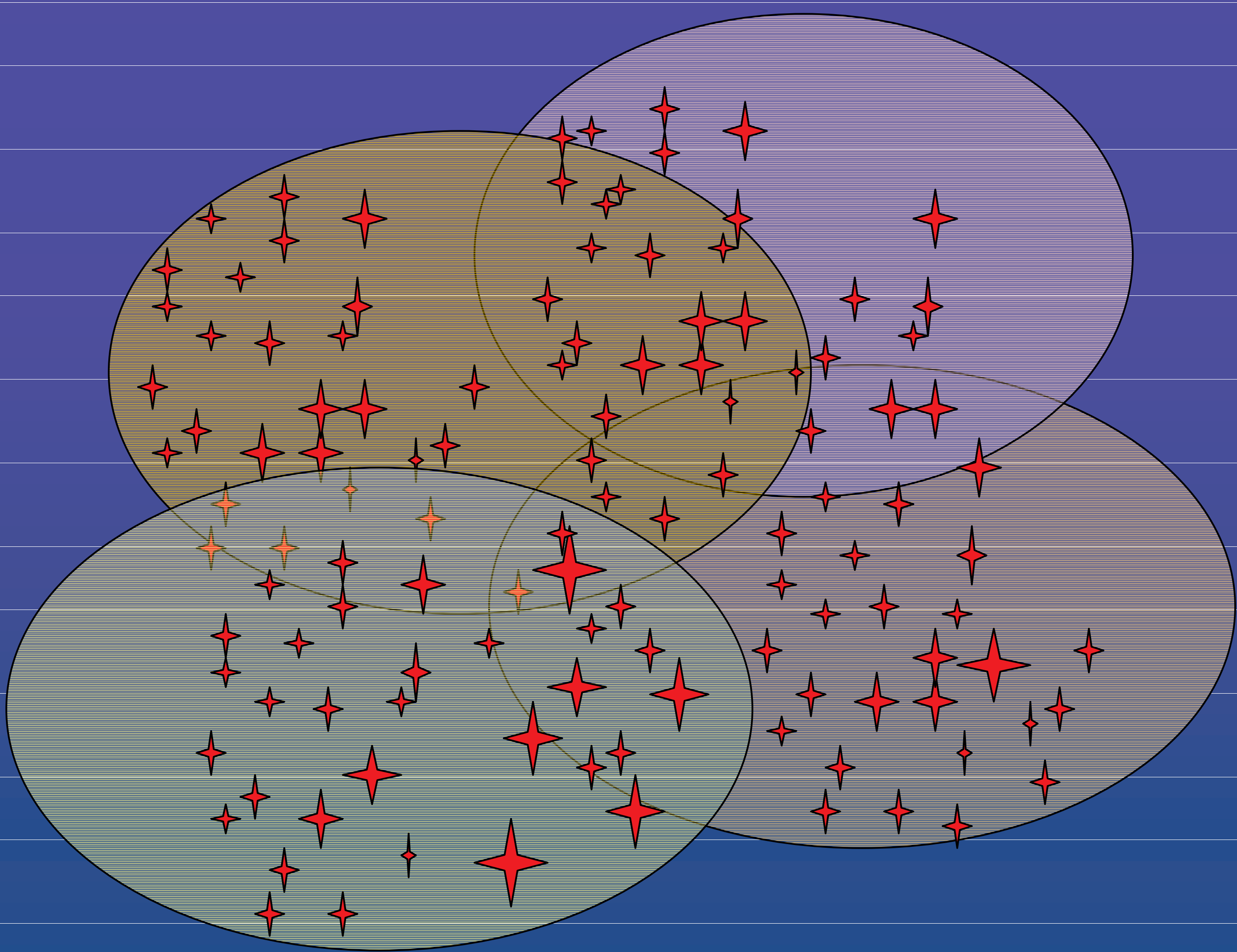
docs(w, cat) = number of documents in the category *cat* containing the word *w*

n = smoothing factor

cats(w) = the number of categories *cats* in which the word *w* occurs

(A similar measure is also used in [Buitelaar & Sacaleanu 2001])

Term Classification



Experiment

- Input: classified documents
- Domains: stock market, management succession and narcotics

Stock Market

- **most relevant terms are nouns**

Aktienbörse 237.05634	[stock-exchange]
Veränderung 143.48146	[change]
Gewinner 142.09517	[winner]
Verlierer 142.09517	[loser]
Hochtief 88.72284	[up and down]
Tief 88.72284	[deep]
Carbon 70.70101	[carbon]
Aktie 53.796547	[stocks]
Kurs 49.768997	[stock price]

Management Succession

- most relevant terms are verbs

berufen 38.45143	[appoint to]
wählen 35.155594	[choose]
übernehmen 32.95837	[accept]
bestellen 28.56392	[nominate]
verlassen 20.873634	[leave]
wechseln 19.77502	[change]
ausscheiden 17.577797	[resign]
nachfolgen 15.380572	[succeed]
zurücktreten 12.084735	[resign]
antreten 8.788898	[assume office]

Learning Relevant Verb Noun Collocations

- **verb noun collocations**

  Ruhestand [retirement], *treten* [step]

  Leitung [leadership], *übernehmen* [take over]

- **German: free word order**

  ⇨ not sufficient to take into account only bigrams, trigrams, e.g., the word order between verbs and nouns is not fixed and very often discontinuous

  ⇨ consider all possible term pairs in a sentence ignoring the linear order

  - verb noun
  - adjective noun
  - noun noun

- **using different association measures (Evert & Krenn, 2000)**

  ⇨ mutual information (Church & Hanks, 1989)

  ⇨T-test (Daille, 1996)

  ⇨ Log-Likelihood (Manning & Schütze, 1999)

Integrating DNLP on Demand

If a sentence contains relevant verbs and verb-noun collocations in addition to other terms, the sentence should be passed to DNLP,

otherwise, SNLP will be sufficient.

Evaluation of Template Extraction

- both at sentential level
- proof of
  - usefulness of hybrid architecture
  - how much improvement deep NLP helps to achieve
  - whether domain modeling gives right predication

Evaluation at Sentential Level: Corpus

- construction of an evaluation corpus based on the following criterion

⇨ the more relevant verbs and nouns a sentence contains, the more relevant is the sentence in this domain.

$$RS = NV * (1 + lg \sum_{i=0}^{NV} RVT_i) + NN * (1 + lg \sum_{j=0}^{NN} RNT_j)$$

where $NV > 0$ and $NN > 0$.

RS: relevance of a sentence
NV: number of relevant verbs in a sentence
NN: number of relevant nouns in a sentence
RVT_i: the relevance of a verb term i occurring in the sentence
RNT_j: the relevance of a noun term j occurring in the sentence

- selection of 50 top relevant sentences in management succession domain

Annotation (manual)

- partially filled templates based on shallow and deep template filling rules
- construction of an idealized case, which is not affected by the performance of the current system

Dr. Heinz Neckar ... appointed ... manager of …, as <u>successor</u> Dr. Herbert Ehrlich retired.

```
<sentence id=1>
 <shallow>
  <template name="management_succession">
     <slot name="person_out">Dr. Herbert Ehrlich </slot>
  </template>
 </shallow>
 <deep>
<template name="management_succession">
   <slot name="person_in">Dr. Heinz Neckar </slot>
</template>
<template name="management_succession">
     <slot name="person_out">Dr. Herbert Ehrlich </slot>
  </template>
</deep>
</sentence>
```

Evaluation at Sentential Level: Scenarios

- applying sentential template merging to
  - shallow templates only
  - deep templates only
  - fusion of shallow and deep templates

Evaluation Results

Shallow		Deep		union(Shallow, Deep)	
coverage	recall	coverage	recall	coverage	recall
0.56	0.31	0.94	0.68	0.96	0.92

coverage= the percentage of sentences, to which the template filler rules can be applied
precision=1.0 (manual annotation)

- verbs play a more important role than nouns and adjectives in management succession domain ⇒ domain modeling

- information extracted by shallow and deep template filling rules is complementary to each other in most cases ⇒ hybrid architecture

Additional Important Issues

- Build rich and robust semantic representation for IE
- Domain specific discourse analysis for template merging
  - Anaphora resolution
  - Ontology inference
  - Evaluation