Natural Language Software Registry
Second edition

Elizabeth Hinkelman Markus Vonerden Christoph Jung

German Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
E-mail: registry@dfki.uni-sb.de

Contents

1 Editor's note **5**

2 Questionnaire **6**

3 Key to Field Names **8**

4 Speech Signal Analysis **10**

3R – Recogniser, Recorder, Reproductive Speech 10
COSIMA 12
CSL – Computerized Speech Lab 13
CSRE – Canadian Speech Research Environment 15
DECtalk 17
Signalyze(TM) 19

5 Morphological Analysis **21**

AMOS 21
AMPLE 23
MORPHIX-3 24
NAUDA generation component * [1] 25
PC–KIMMO 26
TAG–GEN * 27
X2MORF 27

6 Syntactic Analysis **30**

AV parser 30
CFG parser 31
CHARON 32
DISCO chart parser 36
ETL parser 37
GPSG tools 39
GPSG parser 40
JIM III 42
JPSG parser and CU-prolog 43
Linguistic Kernel Processor (LKP) 46
MCHART 47
PAULA 48
PlayMoBild 49
PLEUK 51
SLG 53
UBS – UnifikationsBasierte Sprache 54

[1] Entries marked with a star are simply cross-references to th full entries.

7 Semantic and Pragmatic Analysis **57**
Alvey Natural Language Tools * . . . 57
Context Feature Structure System * . . . 57
NLL . . . 57
SLG * . . . 59
System for evaluation of anaphoric relations . . . 60

8 Generation **62**
AL FRESCO Interactive System * . . . 62
CAT2 * . . . 62
CHARON * . . . 62
DECtalk * . . . 62
ELU * . . . 62
FUF and SURGE . . . 62
GPSG–tools * . . . 65
Linguistic Kernel Processor (LKP) * . . . 65
NAUDA generation component . . . 65
PlayMoBild * . . . 67
STEMMA * . . . 67
TAG–GEN . . . 67
UBS – UnifikationsBasierte Sprache * . . . 69

9 Knowledge representation systems **70**
BACK . . . 70
KRIS – Knowledge Representation and Inference System . . . 71
QDATR . . . 72
RHET . . . 74
TEMPOS . . . 76

10 Multicomponent systems **78**
ALE – Attribute Logic Engine . . . 78
AL FRESCO Interactive System . . . 80
Alvey Natural Language Tools . . . 81
BIM LOQUI . . . 84
CAT2 . . . 86
Context Feature Structure System . . . 88
ELU . . . 91
EVAR, ERNEST . . . 93
Experimental machine translation system . . . 95
KOMET . . . 98
NL Builder 5.0 (TM) . . . 102
NUGGET (R) . . . 103
PAKTUS . . . 107

PARLANCE / Learner . . . 108
PENMAN . . . 110
PLAIN+ . . . 114
PROFGLOT . . . 117
Pundit . . . 119
QPATR . . . 120
SCISOR / NLToolset . . . 122
SNePS . . . 124
SUNDIAL . . . 127
YAKR . . . 129

11 NLP-Tools **131**
COMPULEXIS . . . 131
CUF . . . 132
DCG workbench . . . 133
Dictionary Maintenance Utilities . . . 135
Dictionary Maintenance Programs . . . 137
DITO – DIagnostic TOol for german syntax . . . 140
EGG – editor for GPS-grammars . . . 143
GTU – Grammatik-Test-Umgebung . . . 144
GULP – Graph Unification Logic Programming . . . 145
Linguistic DataBase . . . 146
LINGUIST . . . 148
P–TRA . . . 150
SEMBLEX . . . 151
Term Rewrite System for non-confluent TRS's . . . 153
TFS (Typed Feature Structure) system . . . 155

12 Data sets **158**
FJGH–grammar . . . 158
PC–KIMMO definition files for turkish morphology . . . 160

13 Applications and text processing **162**
ESTEAM (ESPRIT 316) . . . 162
How to Use IT (Mac) . . . 163
How to Use IT (MS-DOS) . . . 164
ILA multilingual toolkit . . . 165
KOREKTOR 2.0 . . . 166
ORFO . . . 167
Parser . . . 169
STAMP . . . 171
STEMMA . . . 172
WORDSURV . . . 173

1 Editor's note

The *Natural Language Software Registry* is a concise summary of the capabilities and sources of language processing software available to researchers. It comprises academic, commercial, and proprietary software, with theory, specifications, and terms on which it can be acquired clearly indicated.

This second edition, containing nearly one hundred software descriptions, owes much to the participants of the 1992 survey of natural language processing software conducted for the *German Ministry for Research and Technology* by *DFKI* and directed by Prof. Wolfgang Wahlster. The Registry now distinguishes among several levels of linguistic analysis, with a full section for natural language generation. It encompasses as well large systems that perform several levels of analysis, linguistic development tools, and selected applications of natural language processing technology.

Because we have relied on developers' reports of system capabilities, estimates of system coverage and robustness are likely to be subjective when they are given at all. We plan to address this in the future by independent reviews of at least the most noteworthy items. Readers are encouraged to notify us of other software they find useful or have developed.

Presently, the Software Registry lists data sets only when the corresponding analysis program is provided. Further sources of information on data sets and other natural language processing resources include:

- the Center for Lexical Research (lexical@nmsu.edu): machine readable dictionaries and processing programs
- the Linguistic Data Consortium (ldc@unagi.cis.upenn.edu): text, tagged text, and spoken language
- the Institute for New Generation Computer Technology (ifs@icot.or.jp): prolog-based symbol processing programs

As we assemble the third edition of the Software Registry, we look forward to cooperation with initiatives and projects of the European Community, such as *ELSNet*, *RELATOR*, and the software survey conducted by the University of Pisa.

The original concept of the Registry is due to Jessie Pinkham. This second edition has been supported by the *German Ministry for Research and Technology* under grant ITW 9002 0 to *DFKI* inc.

Elizabeth Hinkelman

2 Questionnaire

If you have written a system dealing with natural language and if you want to contribute to our registry (which can make your work popular since it is distributed all over the world), you'll either have to fill in the following form or get an electronic version of it to send per e-mail [2].

- Authors:
- Person to contact for software (if different):
- Institution:
- Department:
- Street:
- City/State/Zip:
- Country:
- Phone (with country & area codes):
- Email network & address:
- Name of system:
- Type of system: research system / commercial product / other (specify)
- Primary task of system: linguistic analysis / test of linguistic theory (specify)
  text generation / machine translation / text proofing /
  database interface / other (specify)
- Components (circle all that apply):
  phonological analyzer/generator
  knowledge representation
  morphological analyzer/generator
  discourse structure
  parser/generator
  pragmatic features
  semantic interpreter
  other (specify)
- Components available as independent modules:

(subsequent questions may need a separate answer for each)

- Components can be extended by: the developer / computational linguist /
  linguist / programmer / experienced user / new user
- Data components are: firmly embedded in program / independent of program
- Data provided (count, language, features):
  (Examples: 120,000 entry wordlist for French
  5,000 word LFG lexicon Swahili w/ affixes, English gloss
  15 rule TG for Dutch cross-serial dependencies
  200 node knowledge base for AIDS case histories w/ 10 30-node cases.
- Data components can be extended by: the developer / computational linguist /
  linguist / programmer / experienced user / new user
- Character set used for language data:
  programmable (describe)
  fixed, 16-bit – Unicode
  fixed, 8-bit – ISO (specify, eg ASCII+Latin II)

[2] Addresses: see title page

fixed, 7-bit – ISO (specify, eg US ASCII)
extended ASCII (specify)
other (specify)

- Range of applicable natural languages: (give theoretical or technical limits)
- Approximate number of examples processed successfully, as a power of 10:
- Specify example type: words / sentences / paragraphs / other
- Its coverage level is now: demonstration / small research / large research / production quality / high volume
- Size of system: lines of source code:
  kilobytes of executable:
  years of work:
- Programming language:
- Operating system or hardware:
- Is there a stable version of the system?
- Is there continuing development?
- Summarize the main goals and ideas. Indicate what makes the project a useful and interesting tool for research applications.

- List documents in which the software is described:

  User documentation:

  System documentation:

- Available support: upgrades / source code / consulting / other
- Format for software distribution:
- Price:
- Restrictions on use:

3 Key to Field Names

Software descriptions are organized into chapters, according to the kind of natural language processing they perform. Most correspond to the levels of language analysis that are traditional in linguistics, broadly construed. The exceptions are as follows:

- Generation. Generation programs sometimes fit the categories traditional to linguistic analysis, but not always. Until there is more consensus among researchers as to appropriate categories, the generation programs will be listed in a single chapter, with cross-reference (denoted in the table of contents with an asterisk) to reversible programs.
- Multicomponent systems. These contain modules for several natural language processing functions, which may or may not be extractable. Readers interested in a particular category of analysis may wish to examine this section for relevant software.
- NLP tools. These are auxiliary programs that support computational linguistics research, without performing any of the primary linguistic processing tasks. This includes lexicons. For the second edition of the Registry this also such programs as the Term Rewrite System and TFS, which commit the user to a general formalism rather than a specific linguistic theory. This category will expand and stand on its own in future editions.
- Data Sets. Grammars are included here when they are the input for a processor that is also listed.
- Applications and text processing. This category includes programs which perform linguistic processing which is not directly accessible to the user, as well as text processing and some miscellaneous programs.

Entries are listed alphabetically by program name within these categories.

After the program name is printed its licencing status. Very roughly speaking, some systems are freely available on the understanding that they will be used for research purposes only. Others are commercial software with standard prohibitions on redistribution. A third possibility is the GNU software agreement, which actually mandates redistribution under some circumstances. The Restrictions field spells out the terms of licencing in more detail, relative to this classification. Programs marked Unavailable are included for general informational purposes.

The individual descriptors are explained below.

- Authors: people responsible for program design and implementation.
- Task: primary activity for which the system was designed.
- Description: free text describing the program.
- Components: program: major modules, such as syntactic parser. data: type of data available in the distribution of the program.
- Modularity: program: whether in fact the major modules can be extracted for independent use. data: whether the data is or is not firmly embedded in the program.
- Extensibility: program: whether it is possible to augment the algorithms. data: whether it is possible to augment the data components.

- Size: The number of lines of source code is a rough estimate of the scale of the project. The size of the executable file may be too large for certain computers. The size of the data indicates how much data comes with the distribution of the program.
- Implementation: programming language used.
- Platform: software or hardware required to run the program, such as a particular operating system.
- Languages: languages for which lexicons or grammars are supplied, or to which the algorithms are applicable.
- Retargetability: whether other natural languages can be substituted. Systems can be retargeted at other natural languages if the data components are independent of program code, and if the linguistic theory permits. Linguists tend to be optimistic about the theory.
- Orthography: the character set which the program uses to represent text. Variations on ASCII are the most common, and may be convenient for particular western european languages. EUC is used for several Japanese programs, but only UNICODE is a general standard.
- Examples: Number of examples on which system was tested. This number will refer to words in morphological analysis, sentences in syntactic analysis, and larger units of text for some larger systems.
- Status: whether the project is completed, ongoing, or under development. Eventually, this field will indicate whether there is a stable version of the code, and whether it is being patched when bugs are discovered. It also indicates the coverage level of the system.
- Documentation: manuals or research reports about the system.
- Consulting: whether anyone is available to help users.
- SourceCode: whether the program distribution includes source code.
- Upgrades: whether acquiring the program entitles one to upgrades as well.
- Support: any additional support provided by the software source.
- Format: form in which the program is distributed.
- Price: Even non-profit organizations often pass on distribution costs.
- Restrictions: legal obligations of the user.
- Contact: person who distributes the software.
- Address:
- Telephone:
- Email: Internet electronic mail address.

4 Speech Signal Analysis

3R – Recogniser, Recorder, Reproductive Speech

License: research.
Authors: Joachim Brettschneider.

Task:

- Speaker independent isolated word recogniser.
- Two channel replaying and recording.
- Recorder function.

Description: The 3R software was implemented by Speech Technology Center, Aalborg, Denmark for ESPRIT project 2094 (SUNSTAR). 3R runs on a DSP32C board from LSI.

The main functions of the 3R software are:

- Speaker dependent recognition based on Dynamic Time Warping.
- Speaker independent recognition based on Continuous Hidden Markov Modelling.
- Approximately 50 words active simultaneously (e.g. 20 HMM and 30 DTW or 35 HMM and 15 DTW).
- Word rejection based on signal statistics and garbage models.
- Multichannel replaying and recording of words or messages in parallel to the recogniser.

The training software SIRtrain is used for training the HMM models in the 3R recogniser. Included is also a simulated testing program to check the recognition score after training.

The main functions of the SIRtrain software are:

- Training part.
- Signal Processing Part.
- Testing Part.
- Management Part.

The software is based on the open standard defined by the ESPRIT II project 2589 SAM (Speech Assessment Methodology) and can be used in conjunction with already recorded speech databases in conformance with SAM standards.

Components:

- speech recognition.
- speech players.
- speech recorder.
- training software.

Modularity: 3R software and SIRtrain (training software) available independently.

Extensibility: only extensible by the developer.

Size: -.

Implementation: C, DSP32C assembler (3R), C++ (SIRtrain).

Platform: Loughborough Sound Images DSP32C signal processor board (3R). Sun SPARC, SunOS 4.1.1 (SIRtrain).

Languages: no special language.

Retargetability: all languages applicable, except possibly Greenlandic Eskimo.

Orthography: -.

Examples: ca. 50 isolated words recognized.

Status:

- production quality.
- stable.
- ongoing development.

Documentation: User documentation: 3R Recognizer reference guide, SIRtrain Training Software reference guide.

System documentation: SUNSTAR - internal project documentation.

Upgrades: none.

SourceCode: available.

Consulting: available (telephone).

Format:

- Executable code incl. reference guide.
- Source code incl. complete documentation.

Price: -.

Restrictions: -.

Contact: Joachim Brettschneider, Thomas Renner.

Address: Fraunhofer-Institut für Arbeitswirtschaft und Organisation/ Abteilung 832 / Holzgartenstrasse 17 / 70174 Stuttgart / Germany

Telephone: +49-711-970-2424

Email: brettschneider@iao.fhg.de, renner@iao.fhg.de

COSIMA

License: research.

Authors: Joachim Brettschneider.

Task: Continous speech recognition for the german language.

Description: COSIMA is a research prototype for recognition of continous speech in the German language. COSIMA consists of several components: The signal processing module derives LPC parameters from the original speech signal. The speech recognition module uses Hidden Markov Models of phonemes. The vocabulary to be recognised by the system is specified in a pronounciation lexicon, containing ASCII data. This lexicon can be automatically generated out of the sentences to be recognised. In addition a predictive grammar is obtained to constrain the search path. Speaker independence is achieved by training the phoneme models with speech data from several speakers. Speaker adaptation can be performed by speaking a few example sentences.

Components:

- phoneme lexicon.
- grammar generator.
- signal processing.
- training module.

data:

- stochastic phoneme models (German).
- stochastic grammars for example sentences (German).

Modularity: program: -.

data components independent of program.

Extensibility: program extensible by the developer an the user (speaker adaption).

data extensible by:

- the developer.
- the computational linguist.
- the experienced user (grammars).

Size: -.

Implementation: Fortran, C.

Platform: UNIX SVR3, Sun3.

Languages: German.

Retargetability: none.

Orthography: fixed, 7-bit – ISO (US ASCII).

Examples: words and sentences tested.

Status:

- demonstration.
- stable
- no continuing development.

Documentation: User documentation: COSIMA - Erkennung kontinuierlich gesprochener Sprache, Product Sheet, Fraunhofer Institut für Arbeitswirtschaft und Organisation, Stuttgart, Germany.

System documentation is internal.

Upgrades: none.

SourceCode: none.

Consulting: available.

Format: -.

Price: -.

Restrictions: All code is property of Fraunhofer Institut für Arbeitswirtschaft und Organisation and will only be released under individual license agreements.

Contact: Joachim Brettschneider, Thomas Renner.

Address: Fraunhofer-Institut für Arbeitswirtschaft und Organisation / Abteilung 832 / Holzgartenstrasse 17 / 70174 Stuttgart / Germany

Telephone: +49-711-970-2424

Email: brettschneider@iao.fhg.de, renner@iao.fhg.de

CSL – Computerized Speech Lab

License: commercial.

Authors: Kay Electronics.

Task: speech acquisition, analysis and playback.

Description: The CSL is the most powerful microcomputer-based system available for speech acquisition, analysis and playback. The CSL consists of both hardware and software and works on an IBM PC AT (ISA) or compatible personal computer. The CSL adds user-friendly mouse driven software and high-speed digital signal processing circuitry to an inexpensive PC.

Components: hardware:

- external module (I/O ports, A/D conversion).
- internal card (512 word FIFO memory, two digital signal processing circuits).
- microphone.
- speaker.
- headphones.

software:

- analysis of waveforms.
- FFT and LPC response.
- spectrograms.
- LPC history (pitch synchronous and asynchronous).
- pitch tracking.
- formant frequency (and bandwidth) tracking.
- editing.
- generation of test waveforms.
- IPA character annotation.
- digital filter design package.
- signal playback.

Modularity: embedded.

Extensibility: User can create macros and customize the environment (menus and keystroke sequences). If a user wants to add feature not currently available, it may be wise to discuss the desired capability with Kay because Kay has an ongoing project to add features to CSL.
Presently Kay is not providing the source code for CSL. In its present form it is impractical for a user to alter and recompile the Microsoft C code with new commands. Kay and STR Ltd. are, however, in the process of developing a Programmers Kit to aid in adding new commands. Consult Kay for the delivery and features of this new kit.

Size: 32314 lines of code- 248K compiled (+ 48K for FIR).

Implementation: C.

Platform:

- IBM PC AT compatible with 640K.
- EGA (VGA preferable).
- 20 MB Hard Disk.
- Microsoft mouse.
- 1.2 MB floppy drive.
- 1 free expansion slot.
- 80287 coprocessor.
- ISA bus.

Languages: -.

Retargetability: -.

Orthography: -.

Examples: -.

Status: done/ongoing revision.

Documentation: included with system.

Upgrades: not provided.

SourceCode: not provided.

Consulting: available.

Format: Integrated software/hardware package.

Price: US$5950.00.

Restrictions: contact manufacturer.

Contact: Kay Electronics.

Address: 12 Maple Avenue / P.O.Box 2025 / Pine Brook, NJ 07058-2025 / U.S.A.

Telephone: +1-800-289-5297, +1-201-227-2000 Fax: +1-201-227-7760

Email: -.

CSRE – Canadian Speech Research Environment

License: commercial.

Authors: Donald G. Jamieson, Ketan V. Ramji, Issam Kheirallah, Terry Nearey.

Task: speech analysis/synthesis system.

Description: CSRE (the Canadian Speech Research Environment) is a comprehensive, microcomputer-based system designed to support speech research using IBM/at-compatible micro-computers. CSRE provides a powerful, low-cost facility in support of speech research, using mass-produced and widely-available hardware. The project is non-profit, and relies on the cooperation of researchers at a number of institutions. Version 3.0 Of CSRE has been used since 1989 by researchers in more than 100 laboratories in 12 countries. Version 4.0 offers a wider range of functions, runs faster, uses higher resolution displays, and supports additional hardware systems, including digital signal processing boards. Functions include speech capture, editing, and replay; several alternative spectral analysis procedures, with color and surface/3d displays; parameter extraction/tracking and tools to automate measurement and support data logging; alternative pitch-extraction systems.

parametric speech (Klatt80) and non-speech acoustic synthesis, with a variety of supporting productivity tools.

a comprehensive experiment generator/controller, to support behavioral testing using a variety of common testing protocols.

Components:

- speech editor.
- time-domain analysis.
- spectral analysis.
- formant tracker.
- pitch tracker.
- speech synthesizer.
- acoustic signal synthesizer.
- experiment generator/controller.

data: sample data for tutorials only.

Modularity: program: none.

data firmly embedded in program.

Extensibility: program only extensible by the developer.

data extensible by new or experienced users.

Size:

- 30,000 lines of source code.
- 150 KB of executable.

Implementation: Pascal, C, Assembly.

Platform: DOS on PC-compatible (80386 or higher).

Languages: all acoustic signals.

Retargetability: all acoustic signals.

Orthography: -.

Examples: about 100,000 tokens/words tested.

Status:

- stable.
- ongoing development.

Documentation: Jamieson, D.G., Ramji, K., Kheirallah, I., and Nearey, T.M. (1992) CSRE: The Canadian speech research environment. In J.J. Ohala, T.M. Nearey, B.L. Derwing, M.M. Hodge, and G.E. Wiebe (Eds.) Proceedings of the Second International Conference on Spoken Language Processing, Edmonton: University of Alberta, pp. 1127-1130.

Jamieson, D.G., K. Ramji and T.M. Nearey (1989) CSRE: A Speech Research Environment. Canadian Acoustics, 17, 23-35.

Read, C., Buder, E.H., and Kent, R.D. (1992) Speech Analysis Systems: An Evaluation. Journal of Speech and Hearing Research 35, 314-333.

Read, C., Buder, E.H., and Kent, R.D. (1990) Speech Analysis Systems: A Survey. Journal of Speech and Hearing Research 33, 363-374.

User and System documentation available and supplied with the system.

Upgrades: available.

SourceCode: none.

Consulting: available.

Format: floppy disk.

Price: US$750.

Restrictions: 1 user licence per copy perchased.

Contact: Donald G. Jamieson, Ketan V. Ramji, Issam Kheirallah, Terry Nearey.

Address: University of Western Ontario/ Hearing Health Care Research Unit/ Communicative Disorders / London / Ontario N6G 1H1 / Canada

Telephone: +1-519-661-3901

Email: Jamieson@uwovax.uwo.ca, Ramji@uwovax.uwo.ca

DECtalk

License: commercial.

Authors: Joachim Brettschneider.

Task: Text-to-speech conversion.

Description: The DECtalk system for text-to-speech conversion uses a special synthesizer with cascaded and parallel filter structures. Different parts of the synthesizer are applied for generation of different sounds. Expensive prosodic treatment results in high speech quality. A wide variety of voice parameters can be used to generate certain speaker characteristics. FhG/IAO has developed the German version of the american DECtalk system.

Components:

- parser.
- phonological analyzer.
- morphological analyzer.
- semantic interpreter.
- cascaded/parallel formant synthesizer.

Modularity: none.

Extensibility: program and data extensible by the developer.

Size: -.

Implementation: -.

Platform: proprietary hardware, stand-alone device with V24 interface.

Languages: English, German.

Retargetability: -.

Orthography: fixed, 7-bit – ISO (US ASCII).

Examples: -.

Status:

- demonstration.
- production quality.
- stable.
- no continuing development.

Documentation: User documentation: Ein vollsynthetisches Sprachausgabesystem - DECtalk-, product sheet, Fraunhofer Institut fuer Arbeiswirtschaft und Organisation, Nobelstrasse 12, 7000 Stuttgart 1 Germany.

Upgrades: none.

SourceCode: none.

Consulting: available.

Format: -.

Price: -.

Restrictions: The german version of the DECtalk system was developed by FhG/IAO under contract agreement and can therefore not be made available to the public.

Contact: Dipl.-Ing. J. Brettschneider, Dipl.-Ing. T. Renner.

Address: Fraunhofer-Institut für Arbeitswirtschaft und Organisation / Abteilung 832 / Holzgartenstrasse 17 / 70174 Stuttgart / Germany

Telephone: +49-711-970-2424, +49-711-970-2417

Email: brettschneider@iao.fhg.de, renner@iao.fhg.de

Signalyze(TM)

License: commercial.
Authors: Eric Keller, Ph.D.

Task:

- speech signal analysis.
- pitch extraction.
- manual scoring.

Description: Signalyze is a fast and user-friendly speech analysis system for acoustic signals. Its main features are easy manual scoring, wide, extra wide and narrow band spectrograms in 2 to 256 grays/colors, saveable, fast and reliable pitch extraction, many transformations, splines, envelopes, signal editing: Cut, copy, paste. MacRecorder direct input. File formats: MacRecorder, Impulse, MacAdios, sound resource, numeric (ASCII/TEXT).

Components:

- A/D input.
- spectral analysis.
- pitch extraction, etc.

Modularity: stand-alone shell.

Extensibility: extension via compiled external modules.

Size: Version 1.0: 510K, Version 2.0 (projected) 600K.

Implementation: Think C.

Platform: Macintosh (any Mac from Plus through IIfx).

Languages: All.

Retargetability: -.

Orthography: -.

Examples: 1000-10000.

Status:

- robust or production quality system (used in over 70 labs).
- version 1.0 completed.
- version 2.0 under development (July 1990).
- development ongoing for next few years.

Documentation: CALICO Journal, September 1989; 217- page manual.

Upgrades: provided.

SourceCode: not provided.

Consulting: available (24-hour turnover via email/FAX).

Format: 800k DS floppy disks.

Price: Institutional US$400, Individual US$250.

Restrictions: None.

Contact: Eric Keller, Ph.D. .

Address: University of Quebec at Montreal / Linguistics / C.P. 8888. Succ. A / Montreal, QC H3C 3P8 / Canada

Telephone: no phone, use email please.

Email: 76357.1213@compuserve.com

5 Morphological Analysis

AMOS

License: research.
Authors: Günther Specht.

Task:

- morphosyntactical analysis of ancient Hebrew texts.
- test of linguistic theory and the search for more exact grammar rules, since no native speaker is left.

Description:

- Large texts like the entire Bible.
- Ambiguities in the dictionary due to words not uniquely classifiable at the morphologic level.
- Ambiguities in the syntactic structures as derived.
- Subsumption of the analysed phrases if possible.
- Efficient analysis, also for leftrecursive and quadratic recursiv rules since the bottom-up evaluation strategy is used.
- Note, that the set of resulting tuples is computed all at once! We compute each chapter seperately (not each sentence, like other systems).

Components:

- morphological analyzer.
- morphosyntactical analyzer.
- database interface.

data: knowledge representation as Horn Clauses.

Modularity: SALOMO = morphological analyzer (words) and AMOS = morphosyntactical analyzer (phrases) are available as independent modules.

data: BHT = (Biblia Hebraica Transcripta) text database (the whole Old Testament is available.

Extensibility: Program can be extended by:

- the developer.
- the computational linguist.
- the programmer.

data can be extended by new user by call to the SALOMO-System.

Size:

- 6000 line source code.
- 3,3 MB executable.
- 1 man year of work for kernel.
- 2-3 man year of work for upgrades.

data:

The fact database consists of 25 relations with a total of 117 attributes. One chapter of Genesis contains 167kBytes with 4282 tuples. The entire Genesis has already 7,250 MBytes with 185 897 tuples. To analyze the entire Old Testament we have to handle 108,750 MBytes with 2,7 Mio. tuples. The complete morphosyntactical parser for the ancient Hebrew is represented by about 200 rules (as a logic programm). 4 predicates are linearly recursive (left and right recursion), 13 are mutually recursive and one uses quadratic recursion.

Implementation: Common Lisp (Austin Kyoto Common Lisp).

Platform: UNIX-Workstation, 16MByte Mainmemory.

Languages: ancient Hebrew.

Retargetability: -.

Orthography: ASCII.

Examples: 1991 input half of the Old Testament.

Status:

- production quality.
- high volume.
- stable.
- ongoing development.

Documentation: Specht, G.: Wissensbasierte Analyse althebräischer Morphosyntax; Das Expertensystem AMOS, EOS-Verlag, St.Ottilien, 1990

Specht G.: AMOS - Ein Expertensystem zur morphosyntaktischen Analyse von Althebräisch, Proc. Wissenschaftliches Forum '88, Informationsverarbeitung in Lehre und Forschung, IBM, Muenchen, 1988, pp.63-64

User documentation: AMOS Manual, Version 7, online.

System documentation: Specht, G.: Wissensbasierte Analyse althebräischer Morphosyntax; Das Expertensystem AMOS, EOS-Verlag, St.Ottilien, 1990

Upgrades: available.

SourceCode: available.

Consulting: available.

Format: tar-file on tape.

Price: AMOS is free for universities and research centers.

all others: licence and price on request.

Restrictions: If you don't want to run only the demo, you need the SALOMO-System (Authors: W. Eckardt, G. Specht, same address) for the morphological analysis of the input Text.

Contact: Günther Specht

Address: Technische Universität München / Institut für Informatik / Orleansstr. 34 / 81667 München / Germany

Telephone: +48-89-48095-178

Email: specht@informatik.tu-muenchen.de

AMPLE

License: research.

Authors: David Weber, H. Andrew Black, Stephen R. McConnel.

Task:

- machine translation.
- parsing.

Description: AMPLE is a tool for field linguists working in closely related dialects. AMPLE helps to analyze morphology with the aim of adapting the text to a related dialect. It also can be used as a spelling checker.

Components: morphological analyzer/generator.

Modularity: -.

Extensibility: -.

Size: -.

Implementation: C.

Platform: UNIX, MS-DOS.

Languages: any agglutanive language.

Retargetability: see above.

Orthography: -.

Examples: 10-100.

Status:

- small research.
- ongoing development.

Documentation: AMPLE: A Tool for Exploring Morphology. 252pp.

Upgrades: none.

SourceCode: provided.

Consulting: none.

Format: DOS diskettes.

Price: US$26.00, includes book and diskette.

Restrictions: none, if used for non-commercial purposes & credit given.

Contact: Academic Bookcenter.

Address: Summer Institute of Linguistics / Academic Computing / 7500 W. Camp Wisdom Rd. / Dallas, TX 75236 / U.S.A.

Telephone: +1-214-709-2404

Email: linda@txsil.lonestar.org

MORPHIX-3

License: research, (eventually) commercial.

Authors: Wolfgang Finkler, Günter Neumann.

Task: inflectional analysis and synthesis of written words.

Description: Morphix is an alternative approach to the use of Finite State Automata for morphology in inflectional languages. The essential feature is the use of the morphological regularities of these languages to define a fine-grained word-class-specific subclassification. Morphological analysis and generation can be performed based on this classification by means of simple operations on n-ary trees. The approach handles most of the inflectional phenomena of the German language. In spite of the complexity of the German inflection, the average time required to analyze a word is in the range of CPU-milliseconds (even 0.0002 s on XL1200 Symbolics or on Solborne under Allegro).

Components: morphological analyzer/generator.

data: a stem-lexicon of about 6000 stems classified according to morphosyntactical and morphophonological criteria.

Modularity: The program including a small fullform lexicon can be separated from the files containing expressions to be entered in the lexicon of stems.

data rather independent of program.

Extensibility: extensible by the developer / computational linguist.

data easily extensible since an interactive program queries the user for relevant information.

Size:

- 5251 lines of source code.
- 4 man years of work.

data: 7423 lines of lex-data.

Implementation: standard Common Lisp.

Platform: any that runs Common Lisp.

Languages: German.

Retargetability: none.

Orthography: fixed, 7-bit US ASCII.

Examples: Since the MORPHIX module has already been distributed to more than 20 institutions , there are about 10000 - 100000 expamples tested.

Status:

- stable.
- ongoing development (as reimplementation in C).

Documentation: Wolfgang Finkler and Günter Neumann: MORPHIX - A Fast Realization of a Classification-Based Approach to Morphology. in: Proceedings 4. Österreichische Artificial Intelligence Tagung (OEGAI), Wiener Workshop Wissensbasierte Sprachverarbeitung, Springer Verlag: KI-Fachberichte 176, August 1988, S. 11-19.

System and user documentation: available with the program as LateX-files.

Upgrades: available.

SourceCode: available.

Consulting: none.

Format: requests, please via message to the authors. It's nice to see who is interested in the module. So please don't pass it to other parties, but refer them to the authors!

Price: Common-Lisp version is free.

Restrictions: non-commercial use.

Contact: Wolfgang Finkler.

Address: DFKI Saarbrücken / Stuhlsatzenhausweg 3 / 66123 Saarbrücken / Germany

Telephone: +49-681-302-5269

Email: finkler@dfki.uni-sb.de

NAUDA generation component *

*** see page 65.**

PC–KIMMO

License: research.

Authors: Evan Antworth, Steven McConnel.

Task: morphological parsing.

Description: PC-KIMMO is a new implementation of a program designed to generate and recognize words using Kimmo Koskenniemi's two-level model of word structure. In this model, a word is represented as the correspondence between its lexical form and its surface form. This correspondence is expressed by means of two-level rules. These two-level rules are implemented computationally as finite state transducers.

The PC-KIMMO program is actually a shell program that serves as an interactive user interface around the primitive PC-KIMMO functions. It provides an environment for developing, testing, and debugging two-level descriptions. The primitive functions are also available as a C-language source code library that can be included in a program written by the user.

This means that the user can develop a two-level description using the PC-KIMMO shell program and then link PC-KIMMO's functions into his own program. For example, a syntactic parsing program could use PC-KIMMO as a morphological preprocessor.

Components:

- phonological analyzer/generator.
- morphological analyzer.

Modularity: yes.

Extensibility: They can be extended by modifying the C source code files.

Size: Executable program is 96K.

Implementation: C.

Platform:

- MS-DOS (IBM PC compatible).
- UNIX System V (SCO UNIX V/386 and A/UX) and 4.2 BSD UNIX.
- Apple Macintosh.

Languages: The program is fully general and is intended to be used for any natural language. However, in theory it is possible that some languages, such as Greenlandic Eskimo, might better be modelled as having context-free morphology.

Retargetability: see Languages.

Orthography: -.

Examples: The release software includes sample descriptions of datasets of 10 to 100 words each for nine different languages. Also included is a 20,000-entry lexicon of English.

Status: The primary release of the software is completed. Further extensions and applications are ongoing.

Documentation: PC-KIMMO: a two-level processor for morphological analysis, by Evan L. Antworth, Occasional Publications in Academic Computing No. 16, Dallas, TX: Summer Institute of Linguistics, 1990. ISBN 0-88312-639-7.

Upgrades: provided.

SourceCode: provided.

Consulting: limited consulting available by mail, email, or phone.

Support: Any future upgrades, extensions, or applications will be made available upon request for the cost of media and mailing.

Format: The software on diskette (MS-DOS or Macintosh) is included with the documentation book. The software is also available on various public archives.

Price: US$24.00 for the book including software.

Restrictions: The PC-KIMMO executable program and source code are copyrighted but are made freely available to the general public under the condition that they not be resold or used for commercial purposes.

Contact: Evan Antworth (The published documentation is available from the International Academic Bookstore, same address as follows).

Address: Summer Institute of Linguistics / Academic Computing / 7500 W. Camp Wisdom Road / Dallas, TX 75236 / U.S.A.

Telephone: +1-214-709-2418

Email: evan@txsil.lonestar.org, antworth@am.dallas.sil.org

TAG–GEN *

*** see page 67.**

X2MORF

License: research.
Authors: H. Trost, R. Flassig, H. Pirker.

Task: linguistic analysis.

Description: The system can be split up into two components:

1. The "2-level part" that is responsible for tracking phonological alternations along the lines of standard finite state models.
2. The "word-syntactic" part, modeling morphotactics via a word-grammar that is (in our implementation) oriented towards HPSG.

The system is useful as a morphological front-end for a unification-based natural-language system. As described in (Trost 1991) it can be fully integrated into such a framework. It deviates from standard 2-level morphology in respect to following features:

1. morphotactics is not strictly regular any more.
2. 2-level-rules can be augmented with morphological contexts in form of feature-structures, thus providing an interface between phonology and other levels of linguistic description.

Components: morphological analyzer/generator.

data:

- 2-level rules.
- type type-hierarchy.
- wordlist German.

Modularity: it's one module.

data components independent of program.

Extensibility: extensible by the developer and the programmer.

data easily extensible.

Size:

- 5500 lines of source code.
- 4 man years of work.

data:

- 10 2-level rules (Finite-State-Transducer) handling German orthographic variations.
- 300 type type-hierarchy expressing HPSG-style wordgrammar.
- 500 entry wordlist German.

Implementation: Common-Lisp (Franz Inc. Allegro-Common-Lisp 4.1).

Platform: SPARC SunOS 4.x.

Languages: German.

Retargetability: Two-level models have a strong attitude towards concatenative processes.

Orthography: fixed, 8-bit – ISO 8859.

Examples: about 1000 words tested.

Status:

- demonstration.
- small research.
- stable.
- continuing development.

Documentation: Trost, H. 1990. The Application of Two-Level-Morphology to Non-concatenative German Morphology. In: Proceedings of COLING 90.

Trost, H. 1991. X2MorF: A Morphological Component Based on Augmented Two-Level Morphology. Technical Report RR-91-04, Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken, Germany.

User documentation in preparation.

Upgrades: none.

SourceCode: none.

Consulting: none.

Support: contact.

Format: contact.

Price: contact.

Restrictions: contact.

Contact: Prof. H. Uszkoreit.

Address: DFKI Saarbrücken / Project DISCO / Stuhlsatzenhausweg 3 / 66123 Saarbrücken / Germany

Telephone: +49-681-302-5282

Email: uszkoreit@dfki.uni-sb.de

6 Syntactic Analysis

AV parser

License: research.

Authors: Mark Johnson.

Task: Primary task of system: demo and test of linguistic theory. It is designed to demonstrate unification-based grammars for natural languages.

Description: It provides a general tool for investigating unification-based grammars, with a reasonable amount of graphic support to draw trees and feature structures. These graphics can be cut and pasted into standard Mac word-processing documents.

Components:

- tree- and feature-structure graphics.
- LALR(1) parser-generator (i.e. a "Yacc" in Lisp).

data: 4 demo grammars, based on Shieber's book.

Modularity: The graphics interface, the parser and the data set are available as separate modules.

Extensibility: program and data could be extended by a computational linguist or a programmer.

Size: 2000 line source, 800 K executable.

Implementation: Common Lisp.

Platform: The basic parser runs under any Commonlisp, but the graphics require a Macintosh running MACL 1.3.2. A stand-alone version which will run on a Mac without MACL is also available: send a blank 800k or 1.2Mb diskette to me.

Languages: -.

Retargetability: -.

Orthography: uses Mac character set.

Examples: 1000 sentences.

Status:

- stable.
- ongoing development.
- small research.

Documentation: Readme file.

Upgrades: none.

SourceCode: provided.

Consulting: none.

Format: Binhex'd MACL source files available via email from author, or else send a blank 800k or 1.2Mb Mac diskette and a SASE to me and I will put a standalone parser application on it (MACL is not needed when distributed by diskette).

Price: none.

Restrictions: Restrictions on use: none, although I do request acknowledgement if it is used in other work (e.g. if graphics produced with it appear in publications).

Contact: Mark Johnson.

Address: Institut für maschinelle Sprachverarbeitung - Computerlinguistik / Universität Stuttgart / Keplerstrasse 17 / 70174 Stuttgart 1 / Germany

Telephone: +49-711-121-3132

Email: mj@cs.brown.edu

CFG parser

License: research.

Authors: Rolf Wilkens.

Task:

- linguistic analysis.
- Prooving the ability of a connectionist system to handle the complexity of context-free grammars.

Description: It's not a tool at all. It shows in a simple and lucid manner how a connectionist system works, and that such a network can be used for parsing context-free grammars. By doing this, the systems prooves that it is possible to parse arbitrary context- free grammars with a network.

Components:

- parser.
- generator.

data:

- context-free grammar.
- an input-sentence.

Modularity: program: -.

data independent of the program.

Extensibility: program extensible by the programmer and the developer.

data easily extensible.

Size: -.

Implementation: C.

Platform: Sun Workstation, SunOS 4.x, SunView.

Languages: none.

Retargetability: none.

Orthography: fixed, 7-bit – ISO: US ASCII.

Examples: 10-100 arbitrary CFG with appropriate input tested.

Status:

- demonstration.
- small research.
- stable.
- no continuing development.

Documentation: MA thesis, various publications.

User documentation as Unix manpage.

Upgrades: none.

SourceCode: none.

Consulting: none.

Format: -.

Price: -.

Restrictions: mo warranty.

Contact: Rolf Wilkens.

Address: Sprachwissenschaftliches Institut / Ruhr-Universität Bochum / Universitätsstrasse 150 / 44801 Bochum / Germany

Telephone: +49-234-700-2461

Email: wilkens@ruba.rz.ruhr-uni-bochum.de

CHARON

License: research.

Authors: programming:

Dieter Kohl, Andreas Eisele, Stefan Momma, Jochen Dörre, Markus Geltz.

grammars:

Klaus Netter, Ursula Kärcher, Thilo Tappe, Anette Frank, Judith Meier, Veerle van Geenhoven.

Task:

- linguistic analysis.
- text generation.
- machine translation.
- testbed for parser- and generator-implementations.
- testbed for semantic-components.

Description: The main goal of the charon system is to provide an environment for the development and testing of grammar fragments written in the LFG framework.

The charon system is intended to integrate several originally independent software modules (e.g. the parsers, semantic components and the generator) and to provide a simple general user-interface for the compilation and the test of LFG-grammars. Input and the output of intermediate structures can be redirected to files from within the charon system. The system basically implements the following loop:

- 1) read a sentence or command.
- 2) if it is a command process it and goto step 1.
- 3) analyse the sentence by using the selected parser.
- 4) do something with the resulting f-structure (e.g. select certain parts, use it for a semantic interpretation).
- 5) generate a sentence from the resulting f-structure of step 4.
- 6) goto step 1 by backtracking.

For each step it is possible to display the resulting intermediate structures (or certain parts of the intermediate structures).

Step 4 might consist of several steps, depending on some settings controlled by some commands of the charon system and the concrete call predicate, which activates the system. That means there are different specialised versions of the main loop from above, which make use of different components in step 4.

Although the same grammar source can be used for parsing and generation, the parsers and the generator use their own databases for the internal representation of a grammar. This allows use of the charon-system for sentence-based machine translation as well as for mono-lingual applications.

Components:

- parser.
- generator.
- command interpreter.

Modularity: available as independent modules:

- LFG generator.
- Tomita style parser.
- top-down parser.
- Grammars (LFG) for French, German and translation German to French.

Data components are independent of the program.

Extensibility: System:
extensible by the developer, computational linguist and programmer.

Data:

easily extensible.

Size:

- 22000 lines of source code.
- 500K (Quintus) executable.
- 2300K (Sicstus) executable.

Data:

Translation (EUROTRA):

- French grammar: 121 rules, 2343 word LFG lexicon (fullforms).
- German grammar: 382 rules (wide coverage of phenomena in German syntax), 356 word LFG lexicon (fullform).

Translation (ACORD):

- French grammar: 39 rules, 1032 word LFG lexicon.
- German grammar: 29 rules, 817 word LFG lexicon.

Implementation:

- Prolog (used with Quintus 3.1.1, Sicstus 2.1.6, CProlog).
- additional UNIX shell scripts and makefiles for installation and the treatment of templates in LFG source files.

Platform: UNIX.

Languages: German, French.

Retargetability: epsilon-rules in connection with hidden left-recursive rules in a grammar most probably will result in infinite loops or at least an inacceptable run-time, depending on the parser in use.

The (Sicstus-Prolog) executable for the French grammar requires 5800K. This means, we would expect that a 10000 word lexicon might be the upper limit in the moment.

Orthography: fixed, 7-bit – US ASCII.

Examples: -.

Status:

- small research.
- stable.
- demonstration.
- continuing development.

Documentation: Andreas Eisele, Walter Kasper, Dieter Kohl and Klaus Netter, The Stuttgart LFG System: Its Implementation and Use ACORD Technical Documentation 1.4 November 1989.

Andreas Eisele and Jochen Dörre, A Lexical Functional Grammar System in Prolog Proceedings of the 11th COLING 1986 pp. 551 - 553.

Dieter Kohl and Stefan Momma LFG based Generation in ACORD In: Gabriel Bes The Construction of a Natural Language and Graphic Interface – Results and Perspectives from the ACORD project Springer 1992.

User documentation:

- built-in help system of the charon system.
- A more detailed document is in preparation.

System documentation:

- Readme files and some parts of the source code.
- See list of documents above.

Upgrades: available (via ftp after installation).

SourceCode: available (via ftp after installation).

Consulting: none.

Support: -.

Format: anonymous ftp from ftp@ims.uni-stuttgart.de (as soon as the installation procedure is fixed)

Price: -.

Restrictions:

- Non-commercial use only.
- the tomita-style parser will not work properly with Quintus Prolog 3.1.1, but does work with Sicstus-Prolog, Quintus Prolog 2.4 and CProlog.
- The current version requires some experience to install it on an other machine. A simplified installation is in preparation.

Contact: Dieter Kohl.

Address: Universität Stuttgart/ Institut für maschinelle Sprachverarbeitung/ Azenbergstr. 12 / 70174 Stuttgart / Germany

Telephone: +49-711-121-1350

Email: dieter@ims.uni-stuttgart.de

DISCO chart parser

License: research.

Authors: Bernd Kiefer.

Task: linguistic analysis.

Description: It is a chart parser as a kind of data structure that is fully customizable (in the range of the overall chart parsing idea).

Components: parser.

Modularity: Data components are independent of the program.

Extensibility: The system can be extended by the programmer and the experienced user.

Size:

- 3455 lines of source code.
- 227KB executable.
- ca. 1 man year of work.

Implementation: Franz Allegro Common Lisp.

Platform: SunOS 4.1.1.

Retargetability: -.

Examples: -.

Orthography: any character set possible.

Status:

- stable.
- continuing development.

Documentation: System and User documentation in preparation.

Upgrades: available.

SourceCode: available.

Consulting: none.

Format: -.

Price: free.

Restrictions: ask DFKI.

Contact: Bernd Kiefer.

Address: Deutsches Forschungsinstitut für Künstliche Intelligenz (DFKI)/ Stuhlsatzenhausweg 3/ 66123 Saarbrücken / Germany

Telephone: +49-0681-302-5285

Email: kiefer@dfki.uni-sb.de

ETL parser

License: research.
Authors: Fumio Motoyoshi, Hitoshi Isahara.

Task: linguistic analysis.

Description: -.

Components: parser.

data:

- grammar rules for Japanese inflection..
- dictionary of Japanese auxiliary verb.

Modularity: data components are independent of program.

Extensibility: extensible by the programmer and the developer.

data extensible by

- the developer.
- the computational linguist.
- the linguist.
- the experienced user.

Size:

- 25 KB executable.
- 900 lines of source code.

Implementation: Common Lisp.

Platform: -.

Languages: Japanese.

Retargetability: -.

Orthography: EUC.

Examples: sentences.

Status:

- demonstration.
- small research.
- stable.
- continuing development.

Documentation: none.

Upgrades: none.

SourceCode: available.

Consulting: none.

Format: e-mail and disk.

Price: free.

Restrictions: academic use only.

Contact: Hitoshi Isahara.

Address: Electrotechnical Laboratory / Natural Language Section / 1-1-4 Umezono / Tsukuba / Ibaraki 305 / Japan

Telephone: +81-298-58-5925

Email: isahara@etl.go.jp

GPSG tools

License: research.

Authors: Carla Umbach, Guido Dunker.
Task: machine translation.

Description: Tools to transform external definition-files into an internal format (Prolog-rules).

Each tool includes :

- parser.
- Context analysis (expansion of categories, transitive closures, other consistency-checks).
- generator produces Prolog-rules which are used in the Berlin MU-System (project Kit-Fast).

Components:

- parser.
- generator.

data: several external definition-files for GPSG.

Modularity: available independently:

- GPSG-definition-translator.
- GPSG-FCR-translator.
- GPSG-Alias-translator.
- GPSG-ID-translator.
- GPSG-LP-translator.

data components are independent of program.

Extensibility: program only extensible by the developer.

data extensible by computational linguists.

Size:

- 2000 lines of source code.
- 1 man year of work.

Implementation: Quintus-Prolog.

Platform: UNIX, Sun4

Languages: none.

Retargetability: none.

Orthography: fixed, 7-bit – ISO US ASCII.

Examples: about 10 external files with special syntax tested.

Status:

- small research.
- stable.
- no continuing development.

Documentation: no documents available at the moment.

Upgrades: none.

SourceCode: available.

Consulting: none.

Format: floppy-disk.

Price: free.

Restrictions: none.

Contact: Carla Umbach, Guido Dunker.

Address: TU-Berlin / computer science / Franklinstr. 28-29 / 10587 Berlin / Germany

Telephone: +49-30-314-73126

Email: umbach@cs.tu-berlin.de

GPSG parser

License: research.

Authors: Wilhelm Weisweber.

Task: syntactic analysis and development of a constructive version of GPSG.

Description: The chart parser proceeds bottom-up and is integrated into an experimental machine translation system. It interpretes directly the ID/LP format and the metarules.

Components:

- morphological analyser on the basis of SUTRA.
- parser.

data: German grammar rules and lexicon.

Modularity: data components are independent of program.

Extensibility: program only extensible by the developer.

data components extensible by a computational linguist or linguist who is familiar with GPSG with the help of an editor.

Size: 650 KB executable.

850 KB data:

- 22 main categories, 34 features.
- 22 aliases.
- 76 ID rules.
- 23 LP rules.
- 5 metarules.
- 23 FCRs.
- 265 lexical entries (stem forms).

Implementation: Quintus Prolog 3.1.

Platform: UNIX 4.1, Sun workstation.

Languages: German.

Retargetability: theoretically every natural language.

Orthography: first order terms (Prolog terms).

Examples: about 100 sentences tested.

Status:

- small research.
- stable.
- no continuing development.

Documentation: Ch. Hauenschild, S. Busemann "A constructive Version of GPSG for machine translation" in: E. Steiner, P. Schmidt, C. Zellinsky-Wibbelt (eds.) "From Syntax to Semantics - Insights from Machine Translation" Frances Pinter, London 1988, p. 216-238.

W. Weisweber "Ein Dominanz-Chart-Parser für generalisierte Phrasenstruktur- grammatiken" KIT-Report 45, Institute for Software and Theoretical CS, Technical University of Berlin 1987.

W. Weisweber, S. Preuss "Direct Parsing with Metarules" Procs. Coling-92, Nantes 1992, p. 1111-1115 and extended version in KIT-Report 102, Institute for Software and Theoretical CS, Technical University of Berlin 1992.

User documentation: small (2 pages), a larger one in progress.

System documentation: in progress.

Upgrades: none.

SourceCode: none.

Consulting: available.

Format: 3 1/2" diskettes or ftp.

Price: free.

Restrictions: none.

Contact: Wilhelm Weisweber

Address: Technical University of Berlin / Department for Software and Theoretical Computer Sciences / KIT / Sekr. FR 5-12 / Franklinstr. 28-29 / 10587 Berlin / Germany

Telephone: +49-30-314-24928 / -27778

Email: ww@cs.tu-berlin.de

JIM III

License: research.

Authors: Jim Entwisle.

Task: parsing.

Description: Assignment of lexical roles of words using only inflections and closed-class words as only source of constraints (no lexicon). The parser is set up as a network similar to Small and Reiger's Word Expert Parser. It gives a fairly complete report of sentence structure: locating clauses, prep-phrases (with no attachments), complex NPs (inc. modals and adjectives) and VPs (gerunds, infinitives and auxiliaries).
Returns all possible parses in useful form, ambiguity a specialty. Good for learning/testing English grammar. Envisaged to be part of an NLU system.

Components: parser.

Modularity: single component.

Extensibility: Program is in Scheme. It is envisaged as being one module of a larger system.

Size: 248000 characters, 8000 lines of code.

Implementation: MIT Scheme.

Platform: irrelevant, but UNIX SunOS.

Languages: English.

Retargetability: no.

Orthography: -.

Examples: 100-1000 sentences. 6 sentences from War and Peace.

Status: under development.

Documentation: thesis in progress.

Upgrades: not provided.

SourceCode: can be made available as a tar file.

Consulting: see Contact.

Support: none.

Format: source-code as a tar-file.

Price: unknown.

Restrictions: All licencing agreements should be negotiated through Flinders University.

Contact: Jim Entwisle.

Address: Flinders University / Discipline of Computer Science / Box 2100 / Adelaide, South Australia 5001 / Australia

Telephone: +61-8-275-2874

Email: jim@kurango.cs.flinders.oz.au

JPSG parser and CU-prolog

License: research.
Authors: Hiroshi Tsuda, Koiti Hasida, Hidetosi Sirai.

Task:

- linguistic analysis (JPSG: Japanese Phrase Structure Grammar).
- test of linguistic theory (constraint logic programming language).

Description: CU-Prolog is a symbolic and combinatorial constraint logic programming language especially for constraint-based natural language processing. Most CLP languages treat numerical constraints in terms of algebraic equations or inequations. However, for natural language processing or AI applications in general, symbolic and combinatorial constraints are far more important. CU-Prolog was designed to process the latter kinds of constraint. Compared with Prolog or other constraint logic programming languages, CU-Prolog has the following features.

- CU-Prolog interpreter is written only in the C language.
- CU-Prolog support Partially Specified Term (PST) for feature structures of constraint-based grammar formalisms.
- Its symbolic and combinatorial constraint solver is based on the unfold/fold transformation.

CU-Prolog programs consist of the following Constrained Horn Clauses (CHC), which is an extension of Horn clause. Head :- B1,...,Bn ; C1,...,Cn. The Constraint part (C1,...,Cn) contains user-defined Prolog predicates such as $member(X, [ga, wo, ni])$ or $head - feature - principle(H, M, D)$.

A simple Japanese parser based on JPSG(Japanese Phrase Structure Grammar) is one of the most successful application of CU-Prolog. Various constraints of the constraint-based grammar formalism are naturally and elegantly described with PSTs and combinatorial constraints of CU-Prolog.

Components:

- constraint logic programming language interpreter (CU-Prolog).
- parser (JPSG parser).

data: JPSG word entry.

Modularity: interpreter and parser available independently.

data firmly embedded in the program.

Extensibility: cu-Prolog can be extended by C programmars, JPSG parser can be extended by computational linguists, various users, and Prolog programmars.

data easily extensible.

Size:

- lines of source code: 12,000
- kilobytes of executable: 500KBytes
- man years of work: 3 years

data: 200 word JPSG entry for Japanese.

Implementation: C.

Platform: UNIX, Macintosh, MS-DOS with a DOS Extender.

Languages: Japanese.

Retargetability: suitable for processing constraint-based natural language processing systems.

Orthography: EUC (Macintosh version).

Examples: -.

Status:

- stable.
- small research.
- less continuing development.

Documentation: -.

Upgrades: -.

SourceCode: -.

Consulting: -.

Support: -.

Format: The original UNIX version is registered as an IFS (ICOT Free Software) and available using anonymous FTP from ICOT, Japan. FTP host name: ftp.icot.or.jp, Directory: kbms-clp/unix, File name: cuprolog.tar.Z.

It is ported into Macintosh or MS-DOS (with Dos Extender) by H.Sirai of Chukyo-University. (sirai@csli.stanford.edu, or sirai@sccs.chukyo-u.ac.jp) It is available using anonymous ftp from CSLI, FTP host name: csli.stanford.edu, Directory: pub/MacCup, File name: MacCupE0.78w.sit.hqx (program, Bin Hexed and Stuff It-ed), djcup.lzh (MS-DOS program, running under DOS-extender(386/486 cpu), compressed by LHA), manual.tex (manual for older version of MacCup), sample.p (sample program a la JPSG), util.p (utility program).

Price: -.

Restrictions: -.

Contact: Hiroshi Tsuda, Koiti Hasida, Hidetosi Sirai.

Address: ICOT (Institute for New Generation Computer Technology) / 2nd Lab. / Mita Kokusai Building 21F. / 1-4-28 Mita /Minato-ku / Tokyo 108 / Japan

Telephone: +81-3-3456-3069

Email: tsuda@icot.or.jp, hasida@etl.go.jp, sirai@sccs.chukyo-u.ac.jp

Linguistic Kernel Processor (LKP)

License: research.

Authors: Hans Ulrich Block, Manfred Gehrke, Rudi Hunze, Steffi Schachtl, Ludwig Alois Schmid, Christine Zünkler.

Task: parsing and generating of German.

Description: The system uses Trace and Unification Grammar (TUG), a unification based grammar formalism that has additional rule types for movement rules, allows for unconstrained disjunction of feature equations.

Components:

- parser.
- generator.
- dictionary tool.

data: grammars of German, Japanese and Chinese. German lexicon.

Modularity: no modules available independently.

data components are independent of program.

Extensibility: program extensible by the developer.

data extensible by the computational linguist (grammar) and the experienced user (lexicon tool).

Size:

- 1000 lines of source code.
- 3 MB executable.

data:

- 5,000 word TUG lexicon of German.
- appr. 500 TUG-rules for German.

Implementation: Prolog.

Platform: UNIX.

Languages: German.

Retargetability: -.

Orthography: ASCII.

Examples: -.

Status:

- stable.
- continuing development.

Documentation: Block, H. U. Compiling Trace and Unification Grammar for Parsing and Generation, Proc. ACL Reversible Grammar Workshop, Berkeley 1991.

Block, H. U. and Schachtl, S.: Trace and Unification Grammar, COLING 92, Nantes.

poor user and system documentation.

Upgrades: to be discussed with user.

SourceCode: to be discussed with user.

Consulting: to be discussed with user.

Support: to be discussed with user.

Format: to be discussed with user.

Price: to be discussed with user.

Restrictions: to be discussed with user.

Contact: Hans Ulrich Block.

Address: Siemens AG/ ZFE ST SN 74/ Otto-Hahn-Ring 6/ 81739 München / Germany

Telephone: +49-89-636-44537

Email: block@zfe.siemens.de

MCHART

License: research.

Authors: Henry S. Thompson.

Task: parsing.

Description: Flexible and modular chart-parsing framework. Makes a minimum of assumptions, simply provides a clean basis for constructing chart parsers on top of the basic mechanism. Allows easy experimentation with alternative scheduling strategies, rule invocation strategies and formalisms. Includes exemplary implementations of simple CF-PSG and RTN parsers, both top-down and left-corner.

Components: multi-level agenda; active chart-parsing framework; exemplary parsers.

Modularity: components available as independent modules.

Extensibility: components designed for easy extensibility and customization.

Size: source code 50K.

Implementation: Common Lisp.

Platform: -.

Languages: -.

Retargetability: -.

Orthography: -.

Examples: -.

Status: completed.

Documentation: Thompson, H.S. 1983b. MCHART - A flexible, modular chart parsing framework. In Proceedings of the Third Annual Meeting of the American Association for Artificial Intelligence. AAAI, Stanford, CA.

Upgrades: not provided.

SourceCode: provided.

Consulting: questions answered by email.

Format: source code in ascii text.

Price: none.

Restrictions: any non-commercial.

Contact: Henry S. Thompson.

Address: University of Edinburgh / Human Communication Research Centre / 2 Buccleuch Place / Edinburgh EH8 9LW / Scotland

Telephone: +44-31-667-1011 /-6517

Email: hthompson@uk.ac.edinburgh

PAULA

License: research.

Authors: Sascha W. Felix.

Task:

- linguistic analysis.
- test of linguistic theory.

Description: development of a syntactic parser for German based on the theory of government and binding (GB). The parser is intended to simulate human processing effects and to indentify ungrammatialities.

Components: parser.

Modularity: there's only one component.

data firmly embedded in program.

Extensibility: extensible by a computational linguist or linguist.

Size: -.

Implementation: Modula-2 currently translated into C.

Platform: UNIX/SunOS.

Languages: German.

Retargetability: none.

Orthography: fixed, 7-bit, ASCII.

Examples: sentences.

Status:

- small research.
- stable.
- ongoing development.

Documentation: various papers and reports available from the address above.
User documentation: manual page.
System documentation: technical reports.

Upgrades: none.

SourceCode: none.

Consulting: none.

Format: none.

Price: -.

Restrictions: -.

Contact: Sascha W. Felix.

Address: Lehrstuhl für Linguistik / Universtität Passau / Passau / Germany

Telephone: +49-851-509-181

Email: felix@pille.phil.uni-passau.de

PlayMoBild

License: research.

Authors: Günter Neumann, Gertjan van Noord.

Task:

- bidirectional linguistic deduction.
- parsing and generation of single sentences.
- generation of non-ambiguous sentences.

Description: -.

Components:

- parser.
- generator.

data: grammars for Dutch and German.

Modularity: parser and generator are integrated as a uniform linguistic deduction process.

data components are independent of program.

Extensibility: program extensible by the developer and the Prolog hacker, but there is hardly any documentation available.

data extensible by the developer, (computational) linguists and the experienced user.

Size: -.

data: toy grammar for Dutch, small grammer for German.

Implementation: Prolog.

Platform: needs Quintus compatible Prolog.

Languages: German, Dutch.

Retargetability: Since operations like head wrapping, sequence union etc. can be defined, the grammar allows an elegant treatment of free word-order languages.

Orthography: Grammar is written as Prolog terms, in a PATR-like formalism.

Examples: about 1000 sentences tested.

Status:

- demonstration.
- small research.
- stable.

Documentation: User and system documentation virtually non-existent.

Upgrades: -.

SourceCode: -.

Consulting: -.

Support: -.

Format: -.

Price: free.

Restrictions: -.

Contact: Gregor Erbach.

Address: Universität des Saarlandes/ Computerlinguistik/ 66123 Saarbrücken/ Germany

Telephone: +49-681-302-4117

Email: erbach@coli.uni-sb.de

PLEUK

License: research.
Authors: Jo Calder, Kevin Humphreys, Mike Reape, Chris Brew.

Task:

- linguistic analysis.
- test of various linguistic theories.

Description: Pleuk is intended to be a shell for grammar development, in that many different grammatical formalisms can be embedded within it.

In designing Pleuk, we have attempted to make no assumptions as to the syntax and semantics of grammar formalisms. This means that Pleuk gives relatively little support for the detailed operations of particular grammars—the formalism has to supply parsers and generators. We do provide relatively sophisticated support for manipulating grammars as a whole (in terms of the files that define some grammar), interacting with analysers for those grammars and for the display of grammatical definitions or the results of analysis. The latter is achieved by means of a printer specifically designed for representing information in conventional linguistic terms, e.g. attribute-value diagrams, trees, sets, sequences and arrangements of these.

The following grammatical formalisms currently work with Pleuk:

- Cfg: A simple context-free grammar system, intended for demonstration purposes.
- HPSG-PL: A system for developing HPSG-style grammars, produced at Simon Fraser University, Canada, by Fred Popowich, Sandi Kodric and Carl Vogel.
- Mike: A simple graph-based unification system, enhanced with additional operations for the treatment of free word order proposed by Mike Reape in various publications.
- SLE: A graph-based formalism enhanced with arbitrary relations in the manner of Johnson and Rosner (EACL, 1989) and Doerre and Eisele. Delayed evaluation is used to compute infinite relations. This system has been used for the development of several HPSG-style grammars.
- Term: A term-based unification grammar system, originally developed for the support of Unification Categorial Grammar (Zeevat, Klein and Calder).

Sample grammars are provided for all of these formalisms. Work continues on integrating other formalisms with the system.

Components:

- parsers and generators for various constraint-based formalisms.
- graphical derivation checker.
- relatively high quality printing facilities.

data:

- Medium sized UCG for English.
- Example HPSG grammars in two different formalisms.
- Demo grammar for German Mittelfeld constructions.
- Demo CFG.

Modularity:

- Contains the HPSG-PL System by Popowich, Kodric and Vogel (Simon Fraser University).
- Contains the UCG system (Moens et al, EACL 1989).
- Printers for trees, AVMs and other linguistic diagrams in PostScript and Prolog.

data independent of program.

Extensibility: extensible by the computational linguist and the programmer.

data extensible by the computational linguist and the linguist.

Size:

- 20,000 lines of source code (core system).
- between 400 and 5,000 lines of source code per formalism.
- 2,000-3,000 KB of executable per formalism.

data: Medium sized UCG for English (13 rules, approx. 500 lexical entries)

Implementation: SICStus prolog (2.16 or later).

Platform: SunOS 4.1 .

Languages: English, German.

Retargetability: In principle, no limits; actual limits are very difficult to assess.

Orthography: ISO 8859/1.

Examples: Varies with formalism in use from 1000 (UCG) to 10 (CFG).

Status:

- demonstration.
- small/large research (varies with formalism in use).
- stable.
- continuing development.

Documentation: User documentation: "Pleuk Overview", "Interface", some formalisms (manuals in TeXinfo format, both hard copy and on-line).

System documentation: "Functional backbone" (Core code, same format)

Upgrades: available.

SourceCode: available.

Consulting: available.

Format: compressed, tar file from an FTP server.

Price: free for non-commercial use.

Restrictions: Non-commercial use only; free non-commercial reuse of any code, with appropriate acknowledgements.

Contact: Marc Moens.

Address: University of Edinburgh / Centre for Cognitive Science / 2 Buccleuch Place / Edinburgh EH8 9LW / Scotland

Telephone: -.

Email: pleuk@cogsci.ed.ac.uk

SLG

License: research.

Authors: Esther König.
Task: linguistic analysis.

Description: Grammar interpreter for strictly lexicalized grammars (SLG's) (basic categorial grammar extended by a hypothetical reasoning mechanism). Provides a concise format for defining syntax, basic semantic representations, and syntax-semantics-interface.

Components: parser (implemented), generator (planned).

data: experimental grammar and semantic construction rules.

Modularity: parser and grammar available independently.

Extensibility: extensible by a (computational) linguist or a programmer.

Size: 40 prolog clauses.

data:

- 10 definitions of syntactic categories.
- 20 lexical entries.

Implementation: Prolog, CUF.

Platform: UNIX..

Languages: no special langauge.

Retargetability: all.

Orthography: ASCII.

Examples: 40 phrases.

Status:

- demonstration.
- small research.
- stable.
- ongoing development.

Documentation: none.

Upgrades: none.

SourceCode: none.

Consulting: none.

Format: individual arrangement.

Price: to be negotiated.

Restrictions: non-profit.

Contact: Esther König.

Address: Universität Stuttgart / Institut für maschinelle Sprachverarbeitung (IMS) / Azenbergstrasse 12 / 70174 Stuttgart / Germany

Telephone: -.

Email: esther@ims.uni-stuttgart.de

UBS – UnifikationsBasierte Sprache

License: research.

Authors: Frieder Stolzenburg.

Task:

- linguistic analysis.
- test of linguistic theory (HPSG, Pollard & Sag 1987).

Description: The formalism of HPSG is complex and includes more data structures and operations than earlier grammar foramlisms. In particular there are disjunction, negation, sets, general functions and types that are added to the regular feature structures. These new aspects cannot be implemented in Prolog in a straight forward manner. The system UBS – an extension of SEPIA Prolog and written in SEPIA Prolog itself – has been developed in order to accomodate these aspects. The system follows the tradition of GULP (Covington 1989). GULP is an extension of Prolog that facilitates the specification of feature structures in Prolog programs by allowing order and arity independence. GULP is restricted in that it cannot handle negation, disjunction, sets or functions as feature values. UBS is now able two process typed feature structures over type hierarchy trees, unification, disjunction, (restricted) negation, functional dependent values (relations) and sets (unification of sets). Currently UBS is extended to cope with general type hierarchies as formalized by B. Carpenter (1992).

Components: parser/generator.

data: hpsg grammar for english.

Modularity: it's one module (ubs.pl).

data independent of program.

Extensibility: program extensible by the developer, the programmer and the experienced user.

data easily extensible.

Size:

- 1455 lines of source code.
- 33 KB of executable (+ SEPIA system).
- 2 man years of work.

data: an english grammar.

Implementation: SEPIA (Standard ECRC Prolog Integrating Advanced Features).

Platform: SunOS.

Languages: English.

Retargetability: any languages in principle.

Orthography: Fixed, 7-Bit – ISO.

Examples: about 10 words/sentences tested.

Status:

- demonstration.
- stable.
- continuing development.

Documentation: Stolzenberg, Frieder: UBS – A unification-based language for the implementation of HPSG. (Fachberichte Informatik 2/92) Koblenz: Universität Koblenz-Landau, February 1992.

including System and User documentation: Stolzenberg, Frieder: Typisierte Merkmalsstrukturen und HPSG. Eine Erweiterung von UBS in SEPIA. Diplomarbeit. Universität Koblenz-Landau. November 1992.

Upgrades: none.

SourceCode: available.

Consulting: none.

Format: ASCII, availabel via E-mail from the author

Price: free.

Restrictions: for research only.

Contact: Frieder Stolzenburg.

Address: Universität Koblenz-Landau/ Institut für Informatik/ Rheinau 1/ 56075 Koblenz / Germany/

Telephone: +49-261-9119-426

Email: stolzen@infko.uni-koblenz.de

7 Semantic and Pragmatic Analysis

Alvey Natural Language Tools *

* see page 81.

Context Feature Structure System *

* see page 88.

NLL

License: research.
Authors: John Nerbonne, Joachim Laubsch and Kader Diagne.

Task:

- linguistic analysis.
- interface construction.

Description: NLL is a logical language for representing the meaning of natural language expressions on the computer. Its design has five goals:

- to provide an independent semantics module for NLP, which is realized in very flexible interfaces to NLL, not only through a reader and constructor functions, but also through a compiler from feature structure descriptions.
- to support some semantic inference.
- to provide a base for disambiguation (like SRI's QLF, quasi-logical form) and domain-specific interpretation (e.g., transduction into a database query language, such as SQL, or a language for controlling an application, such as the AKA-MOD agents' system).
- to support experimentation at the level of semantic representation, which is realized through a declarative specification of NLL syntax (and current work is attempting a declarative specification of simplification and transduction to application languages).
- to facilitate the semantic representation of common grammatical constructs (lexical and syntactic), e.g., quantification, plurals, comparatives, questions, predications with variable number of arguments.

Components: semantic interpreter.

data: test files.

Modularity: it's one module, but it is considered separating the language definition tools to allow the definition of alternative semantic representation languages (still quite experimental).

data independent of program.

Extensibility: program extensible by the programmer.

data extensible by the computational linguist or the experienced user.

Size:

- 10 KB of source code.
- 1.24 MB of executable.
- 5 man years of work.

data: Several test files which (i) read NLL expressions; (ii) invoke simplification rules on them; and (iii) convert them to SQL. A test file for converting NLL specifications as might be provided by a feature-based grammar is under construction.

Implementation: Lucid or Allegro Common Lisp, Zebu (Lisp version of YACC).

Platform: UNIX.

Languages: any.

Retargetability: any.

Orthography: N/A.

Examples: about 10000 logical formulas tested (daily experimental use for four years).

Status:

- medium research.
- stable.
- continuing development at both DFKI and University of Groningen.

Documentation: Joachim Laubsch. Logical Form Simplification. STL Report, Hewlett-Packard Laboratories, December 1989.

Abdel Kader Diagne and John Nerbonne. Flexible Semantics Communication in Speech/Language Architectures. In Günter Görz (ed.) KONVENS 92, Berlin: Springer, 348-353.

John Nerbonne, Joachim Laubsch, Abdel Kader Diagne and Stephan Oepen. Natural Language Semantics and Compiler Technology. DFKI Research Report, DFKI-RR-92-50, 1992.

John Nerbonne, Stephan Oepen, Abdel Kader Diagne, Karsten Konrad and Ingo Neis. NLL—Tools for Meaning Representation. In: Stephan Busemann and Karin Harbusch (ed.), DFKI Workshop on Natural Language Systems: Modularity and Re-usability, Saarbrücken, 1993.

John Nerbonne. NLL Models. To appear as DFKI Research Report, 1993.

Joachim Laubsch. The Semantics Application Interface. In Hans Haugeneder (ed.), Applied Natural Language Processing, to appear, 1993.

John Nerbonne. Nominal Comparatives and Generalized Quantifiers. To appear in Jürgen Allgayer (ed.) *Coping with Plurals and Quantifiers: Proceedings of the 1990 German Workshop on Artificial Intelligence Workshop on the Semantics of Plurals and Quantifiers.* Stanford: CSLI, to appear, 1993.

User documentation: An Overview of NLL, Joachim Laubsch and John Nerbonne.

System documentation: in preparation.

Upgrades: available.

SourceCode: available.

Consulting: none.

Format: via ftp.

Price: free

Restrictions: Parts of NLL are copyrighted by the Hewlett-Packard Company, which must be acknowledged in use.

Contact: Kader Diagne

Address: Deutsches Forschungszentrum für künstliche Intelligenz/ (German Artificial Intelligence Center) / Computational Linguistics / Stuhlsatzenhausweg 3 / 66123 Saarbrücken / Germany

Telephone: +49-681-302-5285

Email: diagne@dfki.uni-sb.de

SLG *

*** see page 53.**

System for evaluation of anaphoric relations

License: research.

Authors: Guido Dunker, Carla Umbach.

Task: evaluation of possible antecedents for anaphoric pronouns in texts, developed as a component of an experimental machine translation system.

Description: At the moment the component is part of an experimental machine translation system. Until now factors for possessive and personal pronouns have been developed. It may be extended for the resolution of other anaphoric relations and other factors. It may also be extended for dealing with other sorts of ambiguities.

Components: system for the evaluation of anaphoric relations (represented as first order terms (Prolog terms).

data:

- factors for anaphora resolution.
- weights of the factors.
- application order of the factors.
- 8 factors for German (agreement, binding, proximity, preference for the semantic subject, topic preference, identity of roles, negative preference for free adjuncts, conceptual consistency).

Modularity: none.

Extensibility: The system can be extended by the programmer.

Data components can be extended by a computational linguist who is familiar with the semantic representation FAS.

Size:

- 1860 lines of source code.
- 2 man years of work.

Implementation: Quintus Prolog 3.1.

Platform: UNIX 4.1, Sun Workstation.

Languages: German.

Retargetability: may theoretically be used for every natural language.

Orthography: -.

Examples: about 100 sentences tested.

Status:

- small research.
- stable.
- no continuing development.

Documentation: S. Preuss, B. Schmitz, Ch. Hauenschild, C. Umbach "Anaphora Resolution in Machine Translation" KIT-Report 104, Institute for Software and Theoretical CS, Technical University of Berlin 1993.

G. Dunker, C. Umbach "Verfahren zur Anapherninterpretation in KIT-FAST" KIT-Internal Working Paper in progress, Institute for Software and Theoretical CS, Technical University of Berlin 1993.

B. Schmitz, S. Preuss, Ch. Hauenschild "Textreprsentation und Hintergrundwissen für die Anaphern- resolution im Maschinellen Übersetzungssystem KIT-FAST" KIT-Report 93, Institute for Software and Theoretical CS, Technical University of Berlin 1992 and in: M. Kohrt, Ch. Kueper (eds.) "Probleme der Übersetzungswissenschaft" Working Papers in Linguistics, Department for Linguistics, Technical University of Berlin 1991, p. 39-81.

Ch. Hauenschild "Anapherninterpretation in der Maschinellen Übersetzung" KIT-Report 94, Institute for Software and Theoretical CS, Technical University of Berlin 1992 and Zeitschrift fuer Literaturwissenschaft und Linguistik 84 (1991), Vandenhoeck & Ruprecht, p. 50-66.

System and User documentation in progress.

Upgrades: none.

SourceCode: none.

Consulting: provided.

Format:

- 3 1/2 Inch disks.
- ftp.

Price: free.

Restrictions: none.

Contact: Carla Umbach.

Address: Technical University of Berlin/ Department for Software and Theoretical Computer Sciences/ KIT/ Sekr. FR 5-12/ Franklinstr. 28-29 / 10587 Berlin / Germany

Telephone: +49-30-314-73604 / -27778.

Email: umbach@cs.tu-berlin.de

8 Generation

AL FRESCO Interactive System *

* see page 80.

CAT2 *

* see page 86.

CHARON *

* see page 32.

DECtalk *

* see page 17.

ELU *

* see page 91.

FUF and SURGE

License: research.
Authors: Michael Elhadad.

Task: text generation.

Description: FUF is an extended implementation of the formalism of functional unification grammars (FUGs) introduced by Martin Kay specialized to the task of natural language generation. It adds the following features to the base formalism:

- Types and inheritance.
- Extended control facilities (goal freezing, intelligent backtracking).
- Modular syntax.

These extensions allow the development of large grammars which can be processed efficiently and can be maintained and understood more easily.

SURGE is a large syntactic realization grammar of English written in FUF. SURGE is developed to serve as a "black box" syntactic generation component in a larger generation system that encapsulates a rich knowledge of English syntax. SURGE's input is easy to link to most KL-ONE type of knowledge representation systems (CLASSIC, LOOM, KL-ONE) and the output of SURGE can be easily customized to different situations by setting the value of a small set of feature flags.

SURGE can also be used as a platform for exploration of grammar writing with a generation perspective.

FUF and SURGE are easy to use and learn and have been used in at least 5 occurences in the context of an NLP class, to teach generation.

Components:

- morphological generator (small).
- generator.
- pragmatic features.
- functional unifier with typing and extended control features (FUF).
- linearizer and pattern unifier (for handling ordering constraints).
- lexical chooser (demo version).

data: unification-based grammar of English (SURGE).

Modularity: modules not available independently.

data: SURGE and input forms independent of program.

Extensibility: FUF extensible by the developer/ programmer.

lexical chooser extensible by the new user.
data: SURGE extensible by the computational linguist/ experienced user and the linguist.

Size: FUF: 10000 lines, 300 Kbyte of source code.

data(SURGE): 18000 lines, 550 Kbyte of FUF code.

Implementation: Common Lisp (FUF), FUF (SURGE).

Platform: Any compliant Common Lisp compiler (tested on Lucid Common Lisp, Allegro Common Lisp, Harlequin and CMU Common Lisp on UNIX platforms, Apple Common Lisp on Macintosh, POPL Common Lisp).

Languages: English (SURGE).

Retargetability: Development of unification-based grammars for a new language requires development of new morphological generator (optional) and of a new grammar similar for SURGE. Any language for which unification-based syntactic descriptions are available could be supported if the character set can be supported in Common Lisp.

Orthography: ASCII.

Examples: about 10000 sentences tested.

Status:

- large research.
- stable.
- ongoing development.

Documentation: Most complete description is found in the following dissertation, chapters 3 and 4 - the rest of the dissertation describes other uses of the FUF system (in particular lexical choice): Elhadad, M., "Using argumentation to control lexical choice: a unification-based implementation", Computer Science Department, Columbia University, 1992.

Control features are described in: Elhadad, M. and Robin, J., "Controlling Content Realization with Functional Unification Grammars", Aspects of Automated Natural Language Generation, Springler Verlag, 1992, pp 89-104.

Typing for FUF is described in: Elhadad, M., "Types in Functional Unification Grammars", Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, Detroit, MI, ACL, 1990.

General introduction to FUF and grammar writing in FUF is presented in: McKeown, K. and Elhadad, M., "A Contrastive Evaluation of Functional Unification Grammar for Surface Language Generators: A Case Study in Choice of Connectives", Natutal Language Generation in Artificial Intelligence and Computational Linguistics, Kluwer Academic Publishers, 1991, pp 351-396.

User manual: Elhadad, M., "FUF: The universal unifier - user manual, version 5.0", Columbia University, CUCS-038-91, 1991.

System documentation: documented source code.

Upgrades: available.

SourceCode: available

Consulting: none.

Support: answers to questions by e-mail. The author is interested in getting feedback from users and generally answer to queries within a week.

Format: available by anonymous ftp at as compressed tar file (tar.Z) at:

- cs.columbia.edu:/pub/fuf/fufX.X.tar.Z
- black.bgu.ac.il:/pub/fuf/fufX.X.tar.Z

Latest version is: fuf5.2.tar.Z including the FUF source code, a tutorial including a set of grammars, a set of 500 examples for regression testing the grammar and the user manual in postscript form.

Other material available:

- thesis.ps.Z: compressed postscript version of the dissertation.
- surge2.0.tar.Z: latest version of the SURGE grammar.

Price: free.

Restrictions: GNU type license. The author would like to know who is using the system - so please drop him an e-mail message if you find it useful or make any extensions to either the grammar or the unifier. If you are trying to use the system and it does not perform as you were expecting, please let him know also - He might be able to fix it.

Contact: Michael Elhadad.

Address: Ben Gurion University of the Negev / Computer Science / Beer Sheva 84105 / Israel

Telephone: +972-57-461-626

Email: elhadad@bengus.bgu.ac.il, elhadad@cs.columbia.edu

GPSG–tools *

*** see page 39.**

Linguistic Kernel Processor (LKP) *

*** see page 46.**

NAUDA generation component

License: research.

Authors: Maria Strobel.

Task: sentence generation in the context of a database interface.

Description: The generator produces answers and explanations in the context of natural language database querying. It is used in case of violated presuppositions for rejection and correction, for overanswering and for the natural language presentation of the answer from the database. The generator is able to adapt it's level of granularity to a limited amount to the user's preferences. The core of the component is LanguageAccess' paraphrase generator, which has been extended in order to handle NAUDA's additional capabilities.

Components:

- morphological generator.
- generator.

data:

- application-independent lexica for German.
- application-dependent lexicon for German.

Modularity: LanguageAccess' paraphrase generator independently available.

data components are independent of program.

Extensibility: extensible by the developer and the computational linguist.

data extensible by the developer and the new user.

Size: 1 man year of work.

Implementation: Prolog (VM-Prolog).

Platform: VM.

Languages: German.

Retargetability: other languages not yet implemented.

Orthography: EBCDIC.

Examples: sentences.

Status:

- small research.
- stable.
- no continuing development.

Documentation: Becker, R., D. Kuepper, M. Strobel, D. Roesner: A Cooperative, Natural Language Environmental Information System. In: R. Aiken (ed.): Information Processing 92 - Proceedings of the IFIP 12th World Congress, Vol.2, Elsevier, North-Holland, 1992, pp. 655-665.

Upgrades: none.

SourceCode: none.

Consulting: none.
Format: none.
Price: -.
Restrictions: -.
Contact: Maria Strobel.
Address: FAW / MMK / Helmholtzstr.16 / 89081 Ulm / Germany
Telephone: +49-731-501-460
Email: Bitnet: STROBEL at DULFAW1A, STROBEL at DHDIBM1

PlayMoBild *

*** see page 49.**

STEMMA *

*** see page 172.**

TAG–GEN

License: research.
Authors: Wolfgang Finkler, Karin Harbusch, Anne Kilger.
Task: text generation.
Description: TAG-GEN is a syntactic generator that exploits an incremental and parallel processing scheme. By handling incremental input and producing incremental output, efficiency and flexibility are improved. The goal is to build a system that both runs in real-time and produces output that is highly adaptive to expansions and changes in the input. Incrementality is supported by a distributed, parallel model of active cooperating objects. They verbalize the incremental given input in a lexically guided two-level system, first building the hierarchical structure and then computing the serial order of words in the sentence under construction. Tree Adjoining Grammars are used as syntactic representation formalism and have demonstrated their adequacy in supporting incremental processing.

Components:

- morphological generator.
- generator.

data:

- lexicon.
- grammar.

Modularity: none.

data components independent of program.

Extensibility: only extensible by the developer.

data extensible by the developer and a computational linguist.

Size:

data:

- 100 German lexicon entries.
- 80 English lexicon entries.
- 100 German Tree Adjoining Grammar rules with feature structures.
- 40 English Tree Adjoining Grammar rules with feature structures.

Implementation: Common Lisp, Flavors, CLOS.

Platform: Ivoryboard, Symbolics 36xx, Genera 8.0.1.

Languages: German, English.

Retargetability: all that are principally describable on the basis of Tree Adjoining Grammars.

Orthography: fixed, 7-bit – ISO, ASCII.

Examples: about 100 sentences tested.

Status:

- small research.
- stable.
- ongoing development.

Documentation: K. Harbusch, W. Finkler, A. Schauder. Incremental Syntax Generation with Tree Adjoining Grammars. Proceedings 4. Int. GI-Kongress Wissensbasierte Systeme, 363-374, 23./24. Okt. 1991, Muenchen, Springer Verlag.

A. Schauder. Incremental Syntactic Generation of Natural Language with Tree Adjoining Grammars. DFKI Document D-92-21, German Research Center for Artificial Intelligence (DFKI), Saarbruecken, 1992.

A. Kilger. Realization of Tree Adjoining Grammars with Unification. DFKI Technical Memo, TM-92-08, German Research Center for Artificial Intelligence (DFKI), Saarbruecken, 1992.

J. Bedersdorfer, A. Kilger, M. Weiler. Grammatik und Lexikon fuer TAG-GEN. DFKI Technical Memo, German Research Center for Artificial Intelligence (DFKI), Saarbruecken, 1993.

Upgrades: none.

SourceCode: none.

Consulting: none.

Format: -.

Price: -.

Restrictions: please, contact Wolfgang Finkler and Anne Kilger.

Contact: Wolfgang Finkler, Anne Kilger.

Address: German Research Center for Artificial Intelligence (DFKI) Project WIP Stuhlsatzenhausweg 3 66123 Saarbrücken Germany

Telephone: +49-681-302-5269/ -5271/ -5255

Email: finkler@dfki.uni-sb.de, harbusch@dfki.uni-sb.de, kilger@dfki.uni-sb.de

UBS – UnifikationsBasierte Sprache *

*** see page 54.**

9 Knowledge representation systems

BACK

License: research.

Authors: KIT BACK Group, TU Berlin.

Task: knowledge representation.

Description: The system belongs to the family of Description Logics (aka KL-ONE alike systems, hybrid systems, term subsumption languages, concept logics, terminological logics) which form one of the major paradigms in current research on knowledge representation. Description Logics combine ideas from semantic networks and frames with the formal rigor of first order predicate logic. They support high level conceptual modeling in an object-centered manner.

DL systems have already been used in a number of NLP systems for representing world knowledge or conceptual information. Furthermore, Description Logics are similar to the feature logics underlying unification-based grammar formalisms such as Head-Driven Phrase Structure Grammar.

Components: knowledge representation.

data: none.

Modularity: it's one module.

data: none.

Extensibility: only extensible by the developer.

data: none.

Size:

- 25 000 lines of source code.
- 572 kilobytes of executable.

data: none.

Implementation: Quintus Prolog.

Platform: UNIX.

Languages: none.

Retargetability: none.

Orthography: -.

Examples: -.

Status:

- small research.
- stable.
- ongiong development.

Documentation: User documentation: T. Hoppe, C. Kindermann, J.J. Quantz, A. Schmiedel, M. Fischer, BackV5 Tutorial & Manual, KIT Report 100, Technische Universitaet Berlin, 1993

System documentation: J. Quantz, C. Kindermann, Implementation of the BACK System Version 4, KIT Report 78, Technische Universitaet Berlin, 1990

Upgrades: none.

SourceCode: available.

Consulting: bug reports, recommendations, comments and requests are welcome, but there is no guarantee for any service.

Format: ftp.

Price: -.

Restrictions: only for research purposes.

Contact: Joachim Quantz.

Address: Technische Universitaet Berlin / FB Informatik, FR 5-12 / Franklinstr. 28 -29 / D-10587 Berlin / Germany

Telephone: +49-30-314-254-94

Email: jjq@cs.tu-berlin.de

KRIS – Knowledge Representation and Inference System

License: research.

Authors: Bernhard Hollunder, Armin Laux.

Task: terminological knowledge representation and inference system.

Description: Terminlogical representation system with sound and complete inference algorithms for expressive concept languages. Further extensions such as default reasoning and modal logics are under investigation.

Components: knowledge representation.

Modularity: data components are independent of program.

Extensibility: extensible by:

- the developer.
- the experienced user.

Size:

- 15000 lines of source code.
- 0.4 MByte executable.

Implementation: Allegro Common Lisp.

Platform: Mac, Symbolics, and Sun.

Languages: -.

Retargetability: -.

Orthography: ASCII.

Examples: -.

Status:

- large research.
- stable.
- ongoing development.

Documentation: F. Baader and B. Hollunder, A Terminological Knowledge Representation System with Complete Inference Algorithms, International Workshop on Processing Declarative Knowledge, M. Richter and H. Boley, 567, Springer, 1991.

F. Baader and B. Hollunder, KRIS: Knowledge Representation and Inference System, SIGART Bulletin, 2/3, pages 8-14, 1991.

user and system documentation to appear.

Upgrades: none.

SourceCode: available.

Consulting: none.

Format: ftp.

Price: none for non-profit research.

Restrictions: none.

Contact: Bernhard Hollunder, Armin Laux.

Address: AKA-Tacos / Stuhlsatzenhausweg 3 / 66123 Saarbrücken / Germany

Telephone: +49-681-302-5326, +49-681-302-5327

Email: hollunder@dfki.uni-sb.de, laux@dfki.uni-sb.de

QDATR

License: research, education.

Authors: James Kilbury, Petra Barg, Ingrid Renz, Christof Rumpf.

Task:

- linguistic analysis.
- test of linguistic theory (which make use of non-monotonic inhertit).
- text generation. machine translation. text proofing.
- database interface.
- programming of nonmonotonic inheritance networks.

Description: QDATR is an implementation of the DATR formalism (Gazdar/Evans 1990). DATR is a nonmonotonic inheritance network formalism designed for the representation of natural language lexical information.

Components:

- knowledge representation.
- DATR interpreter.

Example data (i.e. DATR theories) are provided with the system.

Modularity: none.

data components are independent of program.

Extensibility: program extensible by:

- the developer.
- the computational linguist.
- the linguist.
- the programmer.
- the experienced user.

data easily extensible.

Size:

- 69 KB source code.
- 316 KB executable.

Implementation: Prolog.

Platform: DOS.

Languages: DATR theories.

Retargetability: no theoretical or technical limits.

Orthography: fixed, 7-bit – ISO, ASCII.

Examples: -.

Status:

- demonstration.
- small research.
- stable.
- ongoing development.

Documentation: Gazdar, Gerald & Evans, Roger (eds)(1990): The DATR Papers. Cognitive Science Research Paper CRSP 139, School of Cognitive and Computing Sciences, University of Sussex, Brighton.

Kilbury, James; Naerger, Petra; Renz, Ingrid (1992): New Lexical Entries for Unknown Words, Universität Düsseldorf.

System & user documentation: *.DOC files and Gazdar/Evans 1990.

Upgrades: available.

SourceCode: none.

Consulting: available.

Format: disk.

Price: free.

Restrictions: non-commercial only, no further distribution without permission.

Contact: James Kilbury, Petra Barg, Ingrid Renz, Christof Rumpf.

Address: Heinrich-Heine-Universität Düsseldorf / Seminar für Allgemeine Sprachwissenschaft / Universitätsstrasse 1 / 40225 Düsseldorf / Germany

Telephone: +49-211-311-2557

Email: kilbury@ze8.rz.uni-duesseldorf.de, barg@ze8.rz.uni-duesseldorf.de, rumpf@ze8.rz.uni-duesseldorf.de

RHET

License: research.
Authors: Brad Miller, James Allen.

Task: knowledge representation.

Description: This is a Knowledge Representation system based on concepts proved with HORNE. It includes 2 major modes for representing knowledge (as Horn Clauses or as frames), which are interchangeable; a type subsystem for typed and type restricted objects (including variables); E-unification; negation; forward and backward chaining; complete proofs (prove, disprove, find the KB inconsistent, or claim a goal is neither provable nor disprovable); incremental compilation (future); contextual reasoning; truth maintenance; intelligent backtracking; full LISP compatibility (can call or be called by lisp); Allen & Koomen's TEMPOS time interval reasoning subsystem; frames have KL-1 type features, plus arbitrary predicate restrictions on slots within a frame; separate subsystem providing advanced user-interface facilities and ZMACS interface on the lispms.

Components: knowledge representation.

Modularity: firmly embedded.

Extensibility: can be extended under certain conditions.

Size:

- 10MB source.
- dumped image (Allegro 4.1) is about 20meg (with tempos).

Implementation: Allegro Common Lisp 4.1.

Platform: UNIX.

Languages: -.

Retargetability: Any CL; reportedly works under Macintosh CL 2.0.

Orthography: -.

Examples: 10-100.

Status:

- ongoing development.
- stable.
- Replacement system due out in 3q 1993.
- small research.

Documentation: TR 326 (reference) 363 (programmer) 325 (tutorial)

Upgrades: provided.

SourceCode: provided.

Consulting: available.

Support: as available.

Format:

- ftp from cs.rochester.edu:
- /pub/knowledge-tools/* for sources.
- /pub/trs/ai/* for documentation
- or UNIX tar tape.

Price: US$150.00 for tape.

Restrictions: non-commercial w/o negotiated license.

Contact: Brad Miller.

Address: University of Rochester / Department of Computer Science / Rochester, NY 14627-0226 / U.S.A.

Telephone: +1-716-275-1118

Email: miller@cs.rochester.edu

TEMPOS

License: research.

Authors: Johannes A.G.M. Koomen.

Task:

- knowledge representation.
- temporal reasoning.

Description: This is an extension to Rhet to allow reasoning about temporal intervals (Allen's interval logic). It includes the TIMELOGIC temporal reasoning system.

Components: knowledge representation.

Modularity: firmly embedded.

Extensibility: can be extended under certain conditions.

Size: 1.5MB source.

Implementation: Allegro Common Lisp 4.1.

Platform: UNIX.

Languages: Rhet.

Retargetability: Any CL; reportedly works under Macintosh CL 2.0.

Orthography: -.

Examples: 10-100.

Status:

- ongoing development.
- stable.
- small research.

Documentation: TR 231, 307.

Upgrades: provided.

SourceCode: provided.

Consulting: available.

Support: as available.

Format:

- ftp from cs.rochester.edu:/pub/knowledge-tools/*
- or UNIX tar tape.

Price: US$150.00 for tape.

Restrictions: non-commercial w/o negotiated license.

Contact: Brad Miller.

Address: University of Rochester / Department of Computer Science / Rochester, NY 14627-0226 / U.S.A.

Telephone: +1-716-275-1118

Email: miller@cs.rochester.edu

10 Multicomponent systems

ALE – Attribute Logic Engine

License: research.
Authors: Bob Carpenter and Gerald Penn.

Task: general attribute-value grammar formalism.

Description: ALE integrates phrase structure parsing and constraint logic programming with typed feature structures as terms. This generalizes both the feature structures of PATR-II and the terms of Prolog II to allow type inheritance and appropriateness specifications for features and values. Grammars may also interleave unification steps with logic program goal calls (as can be done in DCGs), thus allowing parsing to be interleaved with other system components. While ALE was developed to handle HPSG grammars, it can also execute PATR-II grammars, DCG grammars, Prolog, Prolog-II, and Login programs, etc.

Grammars and logic programs are specified using a typed version of Rounds-Kasper attribute-value logic, which includes variables and full disjunction. Programs are then compiled into low-level Prolog instructions corresponding to the basic operations of the typed Rounds-Kapser logic. There is a strong type discipline enforced on descriptions, allowing many errors to be detected at compile-time.

The logic programming and parsing systems may be used independently or together. Features of the logic programming system include negation, disjunction and cuts. It has last call optimization, but does not perform any argument indexing.

The phrase structure system employs a bottom-up all-paths chart parser. A general lexical rule component is provided, including procedural attachment and general methods for orthographic transformations using pattern matching or Prolog. Empty categories are permitted in the grammar. Both the phrase structure and logic programming components of the system allow parametric macros to be defined and freely employed in descriptions.

Components: Typed Attribute-Value Logic Compiler, comprising:

- inheritance-based object-oriented attribute type system.
- Typed Rounds/Kapser constraint solver (with parametric macros).
- feature structure unification system.
- phrase structure chart-parser with definite clauses.
- lexical rules with orthography.
- definite clause resolver = CLP(Typed Rounds Kasper).

data: small sample categorial and syllabification grammars for English.

Modularity: available as independent modules:

- constraint logic programming system with typed feature structures as terms.
- phrase structure parser.
- type inference, unification, and constraint resolution compilers.

data: the basic data structures, feature structures, are fixed, but incorporate a type system which is user-specifiable.

Extensibility: easy extensible program and data.

Size: 2200 lines of source code

Implementation: Prolog (with first argument indexing and last call optimization).

Platform: any supporting the language.

Languages: English.

Retargetability: any language amenable to phrase structure is applicable.

Orthography: -.

Examples: -.

Status:

- large research (runs about 1000 LI/s on DEC 5100).
- stable.
- continuing development.

Documentation: User documentation:

Bob Carpenter (1992) ALE User's Guide. Carnegie Mellon University Laboratory for Computational Linguistics Technical Report. Pittsburgh.

Upgrades: available.

SourceCode: available.

Consulting: available.

Format: anonymous ftp.

Price: free.

Restrictions: research.

Contact: Bob Carpenter, Gerald Penn.

Address: Carnegie Mellon University / Computational Linguistics Program / Philosophy Department / 135 Baker Hall / Pittsburgh, PA 15213 / U.S.A.

Telephone: +1-412-268-8573

Email: Carp@lcl.cmu.edu, Penn@lcl.cmu.edu

AL FRESCO Interactive System

License: research.

Authors: Olivero Stock.
Task: multimedia Natural Language based information access.

Description: Integration of different components. Natural Language centered system that integrates hypermedia capabilities. Images are retrieved from a videodisc. Use of touch screen; it integrates a knowledge representation system.

Components:

- parser/generator.
- semantic interpreter.
- knowledge representation.
- pragmatic features.

Modularity: some independent modules.

Extensibility: some could be extended; most require quite a lot of inside knowledge.

Size: overall more than 1300 Lisp functions.

Implementation: InterLisp, CommonLisp.

Platform: Sun4.

Languages: Italian.

Retargetability: probably yes.

Orthography: -.

Examples: 100-1000.

Status:

- ongoing development.
- large research.

Documentation: Number of scientific papers, among which:
O. Stock "Parsing with Flexibility, Dynamic Strategies and Idioms in Mind" Computational Linguistics (previously, American Journal of the Association for Computational Linguistics) Vol. 15, N. 2, pp. 1-18, MIT Press, Cambridge, Mass., 1989.

O. Stock, G. Carenini, M. Ponzi, V. Samek Lodovici "Some New Perspectives in Information Access and Interface Design". Proceedings of the IFIP Conference "Modeling the Innovation: Communications, Automation and Information Systems", Noth Holland-Elsevier, 1990.

A. Lavelli, O. Stock "When Something is Missing: Ellipsis, Coordination and the Chart". To appear in Proceedings of COLING90, Helsinki, 1990.

V. Samek Lodovici, C. Strapparava "Integrating Deictic and Linguistics References: The Topic Modulo of the ALFresco System" . To appear in Proceedings of 9th European Conference on Artificial Intelligence ECAI 90, Stockholm, 1990.

E. Franconi, R. Cattoni "Walking Through the Semantics of Frame-Based Description Languages - A Case Study". To appear in Proceedings of 5th International Symposium on Methodologies for Intelligent Systems ISMIS 90, Knoxville, Tennessee (USA), 1990.

User documentation: YAK Manual (for the KR part).

System documentation: AL FRESCO video.

Upgrades: not provided.

SourceCode: not provided.

Consulting: not available.

Support: none.

Format: -.

Price: -.

Restrictions: -.

Contact: Oliviero Stock.

Address: Istituto per la Ricerca Scientifica e Tecnologica / Artificial Intelligence Division / Loc. Pante' di Povo / 38050 Povo (Trento) / Italy

Telephone: +39-461-814-444 Fax: +39-461-810-851

Email: stock@irst.it or stock@irst.uucp

Alvey Natural Language Tools

License: research.

Authors: John Carroll, Clair Grover.

Task: lingiustic analysis.

Description: The MORPHOLOGICAL ANALYSER provides a set of mechanisms for the analysis of complex word forms. The analyser requires data files specifying a lexicon of base morphemes, rules governing spelling changes when concatenating morphemes, and rules describing valid combinations of morphemes in complex words. The tools include a description of English morphology in this form. The analyser should be capable, though, when provided with the necessary linguistic analyses, of being used for most European languages and many others.

There are two alternative PARSERS. The main one is an optimized chart parser, incorporating a 'packing' mechanism (making it much more efficient when parsing sentences containing multiple local ambiguities). The other parser is a non-deterministic LALR(1) parser which seems, in most cases, to be even more efficient than the chart parser.

The GRAMMAR is a wide-coverage syntactic and semantic grammar of English, written in a metagrammatical formalism derived from Generalized Phrase Structure Grammar. Full coverage is provided of the following constructions and their combinations:

- all sentence types: declaratives, imperatives and questions (yes/no, tag and wh questions).
- all unbounded dependency types: topicalisation, relativisation, wh questions.
- a relatively exhaustive treatment of verb and adjective complement types.
- phrasal and prepositional verbs of many complement types.
- passivisation, verb phrase extraposition.
- sentence and verb phrase modification.
- noun phrase complements.
- noun phrase pre- and post-modification.
- partitives.
- coordination of all major category types.
- nominal and adjectival comparatives.

More details and documentation can be obtained by anonymous FTP from ftp.cl.cam.ac.uk (128.232.0.56) in the directory 'nltools/reports'; look at the file 'README' first.

Components:

- morphological analyzer.
- grammar development environment
- parser.
- semantic interpreter.

Modularity: available as independent modules:

- morphological analyzer.
- parser.
- grammar development environment including semantic interpreter.

data components independent independent of program.

Extensibility: program extensible by the Lisp programmer.

data extensible by experienced computational linguist.

Size:

- 37000 lines program source code.
- Wide-coverage grammar of English containing 782 unification-based phrase structure rules.
- 63000 entry English lexicon (40000 homonyms).

Implementation: Common Lisp (conforms to CLtL1).

Platform: Tested on variety of UNIX machines, Apple Macintosh, PC-compatibles. User must supply Common Lisp system. When using data supplied, requires machine at least as powerful as Sun3 with at least 8MB more memory than size of basic Lisp system.

Languages: English.

Retargetability: languages must be non-free word order, segmented at word level.

Orthography: fixed, 7-bit US ASCII.

Examples: about 1000 sentences/ phrases tested.

Status:

- no continuing development.
- large research.
- stable.

Documentation: Russell, G., S. Pulman, G. Ritchie & A. Black, 'A Dictionary and Morphological Analyzer for English', Proc 11th COLING, Bonn, 1986, pp. 277-279.

Briscoe, E., C. Grover, B. Boguraev & J. Carroll, 'A Formalism and Environment for the Development of a Large Grammar of English', Proc 10th IJCAI, Milan, 1987, pp. 703-708.

Ritchie. G., S. Pulman, A. Black & G. Russell, 'A Computational Framework for Lexical Description', Computational Linguistics 13:3, 1987, pp. 290-307.

Ritchie, G., G. Russell, A. Black & S. Pulman, 'Computational Morphology: Practical Mechanisms for the English Lexicon', MIT Press, 1991.

Boguraev, B., J. Carroll, E. Briscoe & C. Grover, 'Software Support for Practical Grammar Development', Proc 12th COLING, Budapest, 1988, pp. 54-58.

User and system documentation:

Ritchie, G., A. Black, S. Pulman & G. Russell, 'The Edinburgh/Cambridge Morphological Analyser and Dictionary System' Software Paper No. 10, Department of AI, University of Edinburgh, 1987.

Carroll, J., E. Briscoe & C. Grover, 'A Development Environment for Large Natural Language Grammars', Technical Report No. 233, Computer Laboratory, University of Cambridge, 1991.

Grover, C., J. Carroll & E. Briscoe, 'The Alvey Natural Language Tools Grammar (4th Release)', Technical Report No. 284, Computer Laboratory, University of Cambridge, 1992.

Upgrades: none.

SourceCode: none.

Consulting: available.

Support: e-mail user group.

Format: anonymous FTP (UNIX tar tape available at extra cost).

Price: 500 ECU (or local equivalent) / 100 ECU for upgrade.

Restrictions: Full 63000-entry lexicon only available for research purposes: commercial use of this lexicon requires negotiation with Longman UK Ltd.

Contact: Sylvia MacKay.

Address: Cambridge / Edinburgh Universities / Lynxvale WCIU Programs / 20 Trumpington Street / Cambridge, CB2 1QA / U.K.

Telephone: +44-223-334-755 Fax: +44-223-332-797

Email: -.

BIM LOQUI

License: commercial.

Authors: BIM-NLteam.

Task: database interface.

Description: Portable NL interface (application dependent parts are isolated from the rest of the system). Dialogue features are regarded as a high priority.

Components:

- morphological analyzer / generator.
- parser / generator.
- semantic interpreter.
- knowledge representation.
- discourse structure.
- pragmatic features.
- logical query evaluation.
- dialogue controller (incl. response determination module).

Modularity: none.

Extensibility: -.

Size:

- 40000 lines of source code.
- 8 MB executable.

Implementation: Prolog by BIM.

Platform: Sun SPARCstation.

Languages: English (prototypes for Dutch and French).

Retargetability: -.

Orthography: -.

Examples: lots.

Status:

- production quality.
- continuing development.

Documentation: -.

Upgrades: provided.

SourceCode: not provided.

Consulting: available.

Format: 1/4 inch tape.

Price: To be negotiated depending on complexity of application.

Restrictions: -.

Contact: Lieve Debille.

Address: sa BIM nv / R&D Kwikstraat, 4 / 3078 Everberg / Belgium

Telephone: +32-2-759-5925

Email: ld@nunbim.be

CAT2

License: research.

Authors: Randall Sharp, Nadia Mesli.
Task: machine translation.

Description: CAT2 is a variant of unification-based formalisms for analyzing, generating and translating natural language. The translation methodology is to transform a syntactico-semantic structure of a source language to a form in which a transfer strategy for translation is optimally simplified (i.e. rules of the form "Haus→house"). The analysis is based on notions within HPSG and GB, i.e. use of Head Features, X-Bar structures, thematic role assignment, binding conditions, adjunct restrictions, etc. The system is easy to learn, and is used for teaching students basic concepts in applied computational linguistics as well as being a production-oriented research and development tool.

Components:

- morphological analyzer/generator.
- parser/generator.
- semantic interpreter.
- translation component.

data:

- linguistic rules for the lexicon.
- linguistic rules for the grammar.
- translation rules for the translation component.

Modularity: rules are independent of the program, written in a user-friendly notation.

Extensibility: rule interpreters are extended by the developer.
linguistic rules can be extended by linguists. translation rules can be extended by translators.

Size:

- lines of source code: 400 different predicates.
- kilobytes of executable: 2000 basic system, 12000 German-to-English system
- 8 man years of work.

data:

- 2000 base entries for German and English (encompassing about 5200 German word forms and 4700 English word forms).
- 10 phrase structure rules for German.
- 20 phrase structure rules for English.
- 200 feature instantiation rules for each of German and English.
- 1800 translation rules from German to English (including disjunctive translations, resulting in about 2000 effective translations).

Implementation: SICStus Prolog.

Platform: UNIX.

Languages: German, English.

Retargetability: Languages tested so far include subsets of: English, German, Spanish, French, Greek, Italian, Dutch, Russian, Japanese. All languages may be simultaneously loaded and tested.

Orthography: fixed, 8-bit – ISO (ASCII as delivered by Sun Workstations).

Examples: about 1000 words and sentences tested.

Status:

- demonstration.
- large research.
- stable.
- ongoing development.

Documentation: Sharp,R. & Streiter,O. (1992) "Simplifying the Complexity of Machine Translation", META 37:4, pp. 681–692.

Sharp,R. (1991) "CAT2: An Experimental Eurotra Alternative", Machine Translation 6, pp. 215–228.

Sharp,R. (1988) "CAT2 – Implementing a Formalism for Multi-Lingual MT", Proceedings of the 2nd International Conference on Theoretical & Methodological Issues in Machine Translation of Natural Languages, June 12–14, Pittsburgh.

user documentation in draft form.

Upgrades: none.

SourceCode: none.

Consulting: available.

Format: runtime system.

Price: free to universities in exchange for 2 high density magnetic cartridges.

Restrictions: For academic research use only, not for commercial purposes.

Contact: Randall Sharp, Nadia Mesli.

Address: IAI / Martin-Luther-Strasse 14 / 6600 Saarbrücken / Germany /

Telephone: +49-681-39313

Email: randy@iai.uni-sb.de, nadia@iai.uni-sb.de, cat2@iai.uni-sb.de

Context Feature Structure System

License: research.

Authors: Martin Böttcher, Michael Koenyves-Toth, Roland Stuckardt.

Task: graph unification formalism.

application domain: unification grammars and knowledge representation, development of grammars and dictionaries, integrated syntactic and semantic text analysis and generation.

currently used for text analysis (integrated parsing constituting five layers of text representation: syntactic structure, thematic structure, reference structure, semantic representation, background knowledge).

Description: The system is a incremental graph unification formalism for recursive, disjunctive and negated attribute graphs.

special efficiency by:

- avoiding copying of repeatedly used parts of graphs.
- distributed disjunctions and virtual agreements.
- independent treatment of logic and graph structure of disjunctions.
- dynamic definition of types.
- inheritance of disjunctive graphs.

text parser:

- efficiently controls unification.
- simultaneously constitutes syntactic and semantic structures and their relationships.

Components:

- morphological analyzer.
- discourse structure.
- text parser.
- semantic interpreter.
- reference resolution.

data:

- dictionary and grammar of German.
- lexicalized text grammar: feature structures which describe the contribution of natural language expressions towards the constitution of sentence structure, thematic structure, reference structure and conceptual structure.
- sentence structure: PLAIN-grammar (P.Hellwig).
- thematic structure: Prague school approach (extensions planned).
- reference structure: referential nets (Ch.Habel) (extensions planned).
- conceptual structure: model of semantic emphasis (J.Kunze), situation semantics.

Modularity: data components are independent of program.

Extensibility: extensible by:

- the developer.
- the computational linguist.
- the linguist.
- the programmer.
- the experienced user.

data:

- see program extensibility.
- the new user.

Size: 700 KB source code.

data: 10,000 dictionary entries for German

Implementation: Common Lisp, Gnu Emacs Lisp, CFS.

Platform: UNIX, Sun.

Languages: German.

Retargetability: languages which can be described with unification grammars.

Orthography: -.

Examples: -.

Status:

- demonstration.
- small research.
- grammar developer's workbench and text parser are stable.

Documentation: [Boettcher/Koenyves-Toth 92] Boettcher, Martin, Koenyves-Toth, Michael: Non-Destructive Unification of Disjunctive Feature Structures by Constraint Sharing. An Abstract Feature Structure Machine. In: Proceedings of the Workshop 'Coping with Linguistic Ambiguity in Typed Feature Formalisms' (ECAI), Vienna, August 1992.

[Haenelt 92] Haenelt, Karin: Towards a Quality Improvement in Machine Translation: Modelling Discourse Structure and Including Discourse Development in the Determination of Translation Equivalents.In: Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-92), June 25-27, 1992, Montreal, Canada.

[Firzlaff/Haenelt 92] Firzlaff, Beate; Haenelt, Karin: Applying Text Linguistic Principles to Modelling Meaning Paraphrases. In: Proceedings of Fifth EURALEX International Congress, University of Tampere, Finland, August 4-9, 1992, pp.213-220.

[Firzlaff/Haenelt 92] Firzlaff, Beate; Haenelt, Karin: On the Acquisition of Conceptual Definitions via Textual Modelling of Meaning Paraphrases. In: Proceedings of the 14th International Conference on Computational Linguistics, July 1992, Nantes, France 1992. pp. 1209-1213.

[Haenelt/Koenyves-Toth 91] Haenelt, Karin; Koenyves-Toth, Michael: The Textual Development of Non-Stereotypic Concepts. In: Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics, April 9-11, 1991, Berlin 1991.

User documentation:

[Boettcher 91] Boettcher, Martin: The CFS System User Manual. A Feature Structure System for Natural Language Analysis Programming. GMD, November 1991.

System documentation: -.

Upgrades: available.

SourceCode: none.

Consulting: none.

Format: -.

Price: -.

Restrictions: -.

Contact: Dr. Karin Hänelt

Address: Gesellschaft für Mathematik und Datenverarbeitung mbH (GMD) / Integrated Publication and Information Systems Institute (IPSI) / Natural Language Systems / Dolivostrasse 15 / 64293 Darmstadt / Germany

Telephone: +49-6151-869-811

Email:

- boettche@darmstadt.gmd.de
- haenelt@darmstadt.gmd.de
- koenyves@darmstadt.gmd.de
- stuckard@darmstadt.gmd.de

ELU

License: research, commercial.

Authors: Amy Winarske.

Task:

- machine translation.
- analysis generation.

Description: ELU (Environnement Linguistique d'Unification) is an enhanced PATR-II style environment for linguistic development written and developed at ISSCO. It is based on unification and its purpose is the development of CL applications.

It provides a declarative environment which allows linguists to write grammars that can be used for both parsing and generation. There is also a transfer component. These three components allow the development of a system which can analyze a text in one language and generate its translation in another.

Components:

- morphological analyzer/generator.
- parser/generator.
- transfer.

Modularity: embedded.

Extensibility: Depends on the module. Generally speaking, extension is difficult.

Size: Depending on architecture and grammar size, 8-16MB.

Implementation: Allegro Common Lisp.

Platform: Sun3, Sun4, SunOS4.

Languages: User defined, Western characters.

Retargetability: Yes. User writes the grammar(s), which include the dictionaries.

Orthography: -.

Examples: 100-1000 sentences (ELU handles words, sentences, and texts).

Status:

- ongoing development.
- large research.

Documentation: Estival, Dominique, "ELU User Manual", Technical Report, Fondazione Dalle Molle, No. 1, Version 1.0, October 1, 1990.

Johnson, R. and M. Rosner (1989) " A rich environment for experimentation with unification grammars", Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics, 182-189.

Russell, G., J. Carroll and S. Warwick (1990) "Multiple Default Inheritance in a Unification-based Lexicon". Proceedings of the First International Workshop on Inheritance in Natural Language Processing, ed. by W. Daelemans and G. Gazdar. ITK, Tilburg, The Netherlands.

Russell, G., S. Warwick and J. Carroll (1990) "Asymmetry in Parsing and Generating with Unification Grammars". Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh.

Upgrades: provided.

SourceCode: not provided.

Consulting: not available.

Format: tape.

Price: 250 Swiss Francs non-profit, 1000 Swiss Francs commercial.

Restrictions:

1. Software may only be used for research and teaching activities by the user.
2. User must cite in all publications or scientific reports involving the software the citation "ELU, Environnement Linguistique d'Unification, developed at ISSCO, University of Geneva".
3. User may not redistribute the software.
4. User recognizes that SUISSETRA and ISSCO do not guarantee the software to be free of errors and are not responsible for maintenance of software.
5. User, in interests of reciprocity, will facilitate the communication of his/her/their work with members of ISSCO who ask for information.

Contact: Institut Dalle Molle pour les Etudes Semantiques et Cognitives (ISSCO).

Address: University of Geneva / 54, route des Acacias / CH-1207 Geneve / Switzerland

Telephone: +41-22-705-7112

Email: elu@divsun.unige.ch

EVAR, ERNEST

License: research.
Authors: Prof. Dr. G. Sagerer, Dr. F. Kummert, Prof. Dr. H. Niemann.

Task: linguistic analysis and dialoque.

Description: The aim of the system EVAR is the automatic understanding of continuous German speech and the handling of an inquiry dialog in the task-domain intercity train connections. To have a structure where linguistic expectations could be used not only for the interpretation but also for the recognition process all kinds of knowledge are integrated in a homogeneous knowledge base. This allows an easy contraint propagation throughout the layers 'dialog', 'pragmatics', 'semantics' and 'syntax'.

The knowledge base is realized by the system ERNEST which provides a framework for the representation of declarative and procedural knowledge based on a suitable definition of a semantic network. The syntax and semantics of the network are clearly defined. In addition, problem-independent rules for the utilization of the knowledge are defined providing both data-driven and model-driven control structures.

By an easy combination of these structures any mixed strategy can be achieved.

Components:

- parser.
- semantic interpreter.
- knowledge representation.
- discourse structure.
- pragmatic features.
- database request.

data: German lexicon.

Modularity: knowledge representation independently available.

data partially independent of program.

Extensibility: program and data extensible by a computational linguist.

Size:

- 160000 lines of source code.
- 14000 KB of executable.
- 20 man years of work.

lexicon: 1100 inflected forms of German.

Implementation: -.

Platform: hardware independent, UNIX-operating system.

Languages: German.

Retargetability: -.

Orthography: programmable in C.

Examples: about 1000 spoken and natural sentences tested.

Status:

- large research.
- stable.
- ongoing development.

Documentation: F. Kummert: Flexible Steuerung eines sprachverstehenden Systems mit homogener Wissensbasis, Bd. 12 von Dissertationen zur künstlichen Intelligenz, Infix, Sankt Augustin, 1992.

G. Sagerer: Automatisches Verstehen gesprochener Sprache, Bd. 74 von Reihe Informatik, Bibiographisches Institut, Mannheim, 1990.

S. Schröder, F. Kummert, G. Sagerer: ERNEST - Die IMMD5 Netzwerk-Umgebung, Technische Unterlage, Lehrstuhl für Informatik 5 (Mustererkennung), Universität Erlangen-Nürnberg, 1989.

Upgrades: available.

SourceCode: available.

Consulting: none.

Format: UNIX tar-file on tape.

Price: negotiable.

Restrictions: none.

Contact: Prof. Dr. G. Sagerer, Dr. F. Kummert (1), Prof. Dr. H. Niemann (2)

Address:

1. Universität Bielefeld Technische Fakultät - AG Angewandte Informatik Postfach 100131, Universitätsstrasse W-4800 Bielefeld 1 Germany
2. Universität Erlangen-Nürnberg Informatik 5 (Mustererkennung) Martensstrasse 3 91058 Erlangen Germany

Telephone: +49-521-106-2935

Email: sagerer@techfak.uni-bielefeld.de

Experimental machine translation system

License: research.

Authors: Christa Hauenschild, Carla Umbach, Birte Schmitz, Susanne Preuss, Wilhelm Weisweber, Stephan Busemann, Guido Dunker, Erich Ziegler, Christian Werner-Meier.

Task: machine translation of texts.

Description: The experimental machine translation system is able to translate texts from German to English sentence by sentence. A module for the evaluation of anaphoric relations of the source language and the knowledge representation system BACK are integrated into the system. The BACK system is used for the representation of the text content in its ABox. The evaluation algorithm uses the representation of the text content in order to check the semantic consistency of possible antecedents for anaphoric pronouns. This factor and others are defined as parameters for the evaluation algorithm. The translation consists of morphological, syntactical, semantical and conceptual analysis, transfer, generation and morphological synthesis.

Components:

- parser/morphological analyser.
- term-rewrite rule interpreter for semantic and conceptual analysis, transfer and generation.
- morphological synthesizer.
- module for the evaluation of anaphoric relations.

data (represented as first order terms (Prolog terms):

- german grammar.
- term-rewrite rules.
- factors for the evaluation of anaphoric relations.

Modularity: components are available independently without morphological synthesizer.

data components are independent of program.

Extensibility: program mainly extensible by the developer and an experienced programmer.

data can be defined by a computational linguist or linguist who is familiar with GPSG, FAS and BACK-System with the help of user interfaces.

Size:

- 650 KB executable.
- 32 man years of work.

linguistic data: 850 KB.

Implementation: Quintus Prolog 3.1.

Platform: UNIX 4.1, Sun workstation .

Languages: German.

Retargetability: theoretically every natural language.

Orthography: -.

Examples: about 100 translation sentences tested.

Status:

- small research.
- stable.
- no continuing development.

Documentation: B. Schmitz, S. Preuss, Ch. Hauenschild "Textrepraesentation und Hintergrundwissen für die Anaphern- resolution im Maschinellen Übersetzungssystem KIT-FAST" KIT-Report 93, Institute for Software and Theoretical CS, Technical University of Berlin 1992 and in: M. Kohrt, Ch. Kueper (eds.) "Probleme der Übersetzungswissenschaft" Working Papers in Linguistics, Department for Linguistics, Technical University of Berlin 1991, p. 39-81.

Ch. Hauenschild "Anapherninterpretation in der Maschinellen Übersetzung" KIT-Report 94, Institute for Software and Theoretical CS, Technical University of Berlin 1992 and Zeitschrift für Literaturwissenschaft und Linguistik 84 (1991), Vandenhoeck & Ruprecht, p. 50-66.

S. Preuss, B. Schmitz, Ch. Hauenschild "Anaphora Resolution Based on Semantic and Conceptual Knowledge" in: S. Preuss, B. Schmitz "Workshop on Textrepresentation and Domain Modelling - Ideas from Linguistics and AI" KIT-Report 97, Institute for Software and Theoretical CS, Technical University of Berlin 1992, p. 1-13.

P. Pause "Zur Modellierung des Übersetzungsprozesses" in: I. Batori, H.-J. Weber (eds.) "Neue Ansätze in Maschineller Sprachübersetzung: Wissensrepraesentation und Textbezug" Niemeyer, Tübingen 1986, p. 45-74.

N. Asher, H. Wada "A Computational Account of Syntactic, Semantic and Discourse Principles for Anaphora Resolution" in: Journal of Semantics 6, 1987, p. 309-344.

W. Weisweber "Transfer in Machine Translation by Non-Confluent Term-Rewrite Systems" Procs. GWAI-89, Eringerfeld 1989, p. 264-269 .

W. Weisweber, Ch. Hauenschild "A Model of Multi-Level Transfer for Machine Translation and Its Partial Realization" KIT-Report 77, Institute for Software and Theoretical CS Technical University of Berlin 1990 and to appear in: Procs. seminar "Computers & Translation '89", Tiflis 1989.

W. Weisweber "Term-Rewriting as a Basis for a Uniform Architecture in Machine Translation" Procs. Coling-92, Nantes 1992, p. 777-783.

Ch. Hauenschild, S. Busemann "A constructive Version of GPSG for machine translation" in: E. Steiner, P. Schmidt, C. Zellinsky-Wibbelt (eds.) "From Syntax to Semantics - Insights from Machine Translation" Frances Pinter, London 1988, p. 216-238.

W. Weisweber "Ein Dominanz-Chart-Parser für generalisierte Phrasenstruktur- grammatiken" KIT-Report 45, Institute for Software and Theoretical CS, Technical University of Berlin 1987.

W. Weisweber, S. Preuss "Direct Parsing with Metarules" Procs. Coling-92, Nantes 1992, p. 1111-1115 and extended version in KIT-Report 102, Institute for Software and Theoretical CS, Technical University of Berlin 1992.

System and user documentation in progress.

Upgrades: none.

SourceCode: none.

Consulting: available.

Format:

- ftp.
- 3 1/2" disk.

Price: free.

Restrictions: none.

Contact: Wilhelm Weisweber.

Address: Technical University of Berlin/ Department for Software and Theoretical Computer Sciences, KIT/ Sekr. FR 5-12/ Franklinstr. 28/29 / 10587 Berlin / Germany

Telephone: +49-30-314-24928 / -27778

Email: ww@cs.tu-berlin.de

KOMET

License: research.

Authors: John Bateman, Elisabeth Maier, Elke Teich, Leo Wanner.

Task: test of linguistic theory (Systemic-functional linguistics) by applying it to the problems of German text generation.

Description: The main goal of the system is to provide a complex component capable of connecting to knowledge of a domain and organizing arbitrarily complex natural German texts in any language style according to set communicative goals. This is attempted within a homogeneous theoretical framework covering all levels of linguistic information relevant for language. The task is broken down into resources at text type, communicative purposes, conceptual ontology, lexicogrammaticization, and semantic interface levels of abstraction. Each of these levels is highly detailed and general. The high degree of stratification considerably simplifies the necessary inter-stratal mappings.

All design decisions are directly motivated by the underlying theory. This theory has inbuilt a very high degree of modularity and open-endedness which has enabled KOMET to be constructed out of the originating Penman system with no code changes, only add-ons, to the original system. No other research NL systems have had this kind of lifespan — work begun on Penman in 1980 — being worked upon by several generations of research students. The life of the projects is directly attributable to the 'broad base' approach to linguistic description. Of principle value currently is the ease of extension being shown to deal with issues of context and pragmatics in text planning, ease of extension to deal with multiple languages, and the breadth of the resources upon which one can build in adding new functionalities and coverage.

Components:

- morphological generator.
- sentence generator.
- text planner.
- knowledge representation.
- discourse structure.
- pragmatic features.

data:

- systemic networks for German grammar.
- systemoc networks for discourse relations.
- closed class lexicon.
- semantic features.
- linguistically motivated ontology ("upper model").

Modularity: Except for the knowledge representation, all of the above components are realized by the same component (the systemic network interpreter of the Penman system extended for text level networks) running off different data (grammars, semantics). The knowledge representation system used is Loom. In addition, data components are also highly modularized and can be considered independently if required (although they typically expect a very rich environment in order to function).

data components are independent.

Extensibility: program extensible by:

- the developer.
- the computational linguist.
- the new user (modifications have been made by users with less than a month's experience with the system).

data:

- the developer.
- the computational linguist.
- the linguist.
- the new user.

Size: about 13MB executable.

data:

- 500 system (disjunctions) systemic network for German grammar (a systemic network disjunction is slightly reminiscent of a GPSG metarule: the network thus has considerable coverage).
- 500 word closed class lexicon for German.
- 500 semantic distinctive features mapping from ontology to grammar.
- 200 semantic distinctive features for mapping from situations to lexicogrammatically commited configurations.
- 250 concept purely linguistically motivated ontology.
- 100 system systemic network for discourse relations.

Implementation: Common Lisp.

Platform: anything that runs Common Lisp.

Languages: German.

Retargetability: No theoretical or technical limits known. This system covers German as the Penman system covers English. It also form part of the experimental multilingual system ML-PENMAN, which has ongoing fragments being written for Dutch, Chinese, Japanese, and French. (For information: contact bateman@gmd.de)

Orthography: standard ASCII.

Examples: about 1000 sentences tested.

Status:

- large research.
- stable.
- ongoing development.

Documentation: John A. Bateman, Maier, Elisabeth A., Elke Teich and Leo Wanner (1991) "Towards an Architecture for Situated Text Generation", In: Proceedings of the International Conference on Current Issues in Computational Linguistics, Penang, Malaysia. (Also available as technical report of GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany.)

Elke Teich (1991) "The KOMET grammar of German." Technical report of GMD/Institut fuer Integrierte Publikations- und Informationssysteme, Darmstadt, Germany.

Maier, Elisabeth A. and Eduard H. Hovy (1991) "A Metafunctionally Motivated Taxonomy of Discourse Structure Relations", In: Proceedings of the 3rd European Natural Language Generation Workshop, Innsbruck, March, 1991. (Also available as technical report of GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany.)

Leo Wanner and Elisabeth A. Maier (1991) "Lexical Choice as an Integrated Component of Situated Text Planning", In: Proceedings of the 3rd European Natural Language Generation Workshop, Innsbruck, March, 1991. (Also available as technical report of GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany.)

User documentation: The Nigel Manual (ISI, Penman system) is still the documentation for the grammar development environment.

System documentation: Lexicalization and Text Planning: forthcoming PhD dissertations from Wanner and Maier respectively.

Upgrades: available.

SourceCode: available.

Consulting: available.

Format:

- unix.
- tar.
- ftp.

Price: currently free.

Restrictions: licenses must be signed with IPSI for the KOMET grammar, and with ISI for the underlying Penman and Loom systems.

Contact: John Bateman.

Address: GMD/IPSI KOMET text generation department Dolivostr. 15 64293 Darmstadt Germany

Telephone: +49-6151-869-826

Email: bateman@gmd.de

NL Builder 5.0 (TM)

License: commercial.
Authors: Edwin R. Addison.

Task:

- NLP shell for interfaces.
- text processing.
- machine translation.
- education.
- research.

Description: NLP developer's Workbench, user can develop NLP application or experiment.

Components:

- tokenizer.
- dictionary.
- morphological analysis.
- parser.
- semantic interpretation.
- semantic network KRL.
- lexical acquisition tools.
- "C" hooks.
- trace and debug software.
- English to SQL.

Modularity:

- available as firmly embedded modules.
- individual components available by arrangement.

Extensibility: -.
Size: 40,000 lines of C Code or 500K exe file, dictionary may grow arbitrarily.
Implementation: C.
Platform: PC, Mac, Apollo, Sun, VAX, NeXT, others.
Languages: English or other ASCII based languages.
Retargetability: may be substituted via grammar/lexicon.

Orthography: -.

Examples: number of systems: 1000+.

Status: commercially available.

Documentation: NL Builder User's Manual.

Upgrades: provided.

SourceCode: available (US$100,000.00).

Consulting: available.

Support: maintenance contract, training course.

Format: floppy or tape, depending on platform.

Price: PC or Mac Version: US$395.00; NeXT US$795.00; Apollo US$2,495.00; Sun US$2,995.00; VAX US$4,995.00.

Restrictions: single user for development; multiple user or reseller licenses available.

Contact: Edwin R. Addison.

Address: Synchronetics, Inc. / Synchronetics, Inc. / 301 N. Front St. / Baltimore, MD 21202 / U.S.A.

Telephone: +1-301-752-1065

Email: -.

NUGGET (R)

License: commercial.

Authors: Konrad Jablonski, Armin Rau, Johannes Ritzke.

Task:

- text generation for XPS (a commercial product).
- prototype integration in a complete database interface and in a UNIX help system (the other system parts e.g. the parser are research prototypes of the ESPRIT P311/ADKMS project).

Description: Speech act theory based text generation system which uses the semantic & pragmatic information (coded in a developed formalism) to restrict the generation of possible sentences out of the large volume of DCG rules. The system is text oriented and uses pronouns where possible. It is prepared to communicate with a parser (common discourse data). The system is a commercial product and well tested in practical use. The system communicates in a well defined language with application programs (XPS, databases, UNIX help systems - all tested).

Components:

- knowledge representation.
- morphological analyzer/generator .
- discourse structure.
- parser/generator.
- pragmatic features.
- semantic interpreter .
- text heuristics.

data:

- German (commercial version) lexicon, DCG rules and morphological rules.
- Italian (prototype version).

Modularity: available independently:

- DCG.
- lexicon devel. system.
- lexicon.
- generation system.

data components are independent of program.

Extensibility: extensible by:

- the developer.
- the computational linguist.
- the linguist.
- the programmer.

data extensible by:

- the developer.
- the computational linguist.
- the linguist.
- the programmer.

Size:

- 7200 lines of source code.
- 870 lines of lex.dev.system.
- 260 KB prolog source code
- 290 KB executable.
- 6 man years of work.

data:

- 600 lexicon entries.
- 220 DCG rules.
- morphological rules.

Implementation: Prolog (IF-Prolog, SNI-Prolog or QUINTUS) .

Platform: UNIX (TARGON OS, SINIX, SunOS), DOS .

Languages: German, Italian.

Retargetability: -.

Orthography: fixed, 8-bit – proprietary ASCII.

Examples: in commercial use.

Status:

- production quality.
- stable.
- ongoing development.

Documentation: K.Jablonski, A.Rau, J.Ritzke: "Konzeption und Architektur des taktischen Textgenerierungssystems NUGGET", WISBER-Memo 12; Saarbruecken Mai 1987.

K.Jablonski, A.Rau, J.Ritzke: Syntax- und Morphologieverarbeitung im Textgenerierungssystem NUGGET. in: Abstracts der Jahrestagung der Deutschen Gesellschaft für Sprachwissenschaft, Wuppertal März 1988; auch in: WISBER-Arbeitsunterlage U 21.

K.Jablonski, A.Rau, J.Ritzke: Anwendung einer logischen Grammatik zur Generierung deutscher Texte. in: H.Trost (Hrsg.): 4. Österreichische Artificial-Intelligence-Tagung, Wien, August 88, Proceedings; Berlin etc.: Springer 1988; S. 83-93; auch als WISBER-Bericht 25, Juni 1988.

K.Jablonski, A.Rau, J.Ritzke: NUGGET - Ein DCG-basiertes Textgenerierungssystem. WISBER-BERICHT 27, August 1988.

K.Jablonski, A.Rau, J.Ritzke: Benutzerfreundlichkeit durch natürliche Sprache: NUGGET - ein Textgenerierungssystem. in: S.Savory (Hrsg.): Expertensysteme: Nutzen für Ihr Unternehmen; München-Wien: Oldenbourg, 2. überarbeitete u. erweiterte Aufl. 1989, S. 365 - 392.

H.U.Block, B.Frederking, M.Gehrke, H.Haugeneder, R.Hunze, K.Jablonski, A.Rau, J.Ritzke, S.Schachtl: Sprachanalyse und Textgenerierung im natürlich-sprachlichen Beratungssystem WISBER; in: W.Brauer/C.Freksa: Wissensbasierte Systeme, 3. Internationaler GI-Kongress München 1989, Berlin, Heidelberg etc.: Springer 1989; S. 275 - 285.

U.Bauernfeind: Natürlich-sprachliche Textgenerierung macht Expertensystem-Schlussfolgerungen durchschaubarer. in: unix/mail (7) 1989; Heft 4; S. 38-42.

K.Jablonski, A.Rau, J.Ritzke: Wissensbasierte Textgenerierung - Linguistische Grundlagen und softwaretechnische Realisierung; Tübingen: Narr 1990.

K.Lehner: Wissensbasierte Lehrsysteme; München: Oldenbourg 1990.

K.Jablonski, J.Samuel: Natural Language Information Access System; in: ESPRIT - Information Processing Systems; Bruessel 1990.

User documentation: K.Jablonski, A.Rau, J.Ritzke: Wissensbasierte Textgenerierung - Linguistische Grundlagen und softwaretechnische Realisierung; Tübingen: Narr 1990.

System documentation is internal.

Upgrades: available.

SourceCode: available.

Consulting: available.

Support: application projects.

Format:

- streamer tapes.
- floppy disks.

Price: executable code as separate module included in TWAICE (use the current SNI price list for the different hw platforms). prices for source code, consulting, application projects on demand.

Restrictions: the lex.dev.systems uses SNI's softw.dev.envir. XSD/E for the user interface .

Contact: Konrad Jablonski.

Address: SIETEC Consulting GmbH / Business unit of SIEMENS NIXDORF Informationssysteme AG / Projekt-Zentrum / Expertensysteme & KI / Riemekestr. 160 / 33106 Paderborn / Germany

Telephone: +49-5251-8-31863 Fax: -31869

Email: jablonski.pad@sni.de

PAKTUS

License: commercial.
Authors: Bruce Loatman.
Task:

- parsing.
- understanding.

Description: PAKTUS (PRC Adaptive Knowledge-based Text Understanding System):

- broad coverage of core English grammar and lexicon.
- easy adaptability to sublanguage grammar and lexicon due to its being a shell.
- easy to use graphical programming interface for development.
- able to handle real scientific and other text.
- reliable information extraction.

Components:

- morphological analyzer.
- parser.
- semantic interpreter.
- knowledge representation.
- discourse structure.
- database / knowledge base generator.
- message router.

Modularity: modules not completely independent, but loosely coupled.
Extensibility: easy.
Size:

- over 275 arcs in the ATN, over 11,000 lexical items.
- over 2 megabytes of Lisp code and structures.

Implementation: Common Lisp.
Platform: Macintosh II, Sun.
Languages: English, Spanish.
Retargetability: relatively easily, depending on similarity of grammar to that of English.
Orthography: -.

Examples: 1000-10000 messages or news reports.

Status:

- ongoing development.
- stable.
- production quality.

Documentation: Loatman, B.; Yang, C-K.; Post, S.; and Hermansen, J. Natural Language Understanding System. Patent Number 4-914-590, U.S. Patent and Trademark Office, April 3, 1990.

Loatman, B. and Yang, C-K. Interactive Graphic Natural Language Programming System. Patent Application 4028-24, U.S. Patent and Trademark Office, pending.

Loatman, B. 1987. "A Hybrid Architecture for Natural Language Understanding." Proceedings of Applications of Artificial Intelligence V, SPIE. Orlando, FL, May 1987.

Upgrades: provided.

SourceCode: not provided.

Consulting: available.

Format: Sun tar, Macintosh floppies.

Price: none.

Restrictions: licensing agreement for clients and subcontractors only.

Contact: Bruce Loatman.

Address: Planning Research Corporation / Government Information Systems / Technology Division / 1500 Planning Research Drive / McLean, Virginia 22102 / U.S.A.

Telephone: +1-703-556-1646

Email: prcrs!loatman@uunet.uu.net

PARLANCE / Learner

License: commercial.
Task: database interface (Learner is knowledge acquisition system for Parlance).

Description: Parlance separates the issue of language understanding from database command generation. Using a rigorous linguistic model of the structure of language and a principled approach to the representation of meaning, the Parlance software translates English into a logical form before generating database commands. This provides a mechanism for detecting contradictions and ambiguities and for understanding the query in context.

The Parlance system handles a richer variety of ambiguous English than any other commercial natural language interface. When it detects multiple interpretations of a query, it presents paraphrases of those interpretations for the user to choose among.

In addition to its language understanding capabilities, Parlance has a graphical human interface featuring interactive help, word definition facilities, and gateways to spreadsheet and graphics systems.

Parlance has been used with a variety of databases, including trading, banking, inventory, and personnel, with vocabularies ranging from a few hundred words to about ten thousand. The system is designed to be adapted to virtually any relational database application.

Components: translator to SQL for database interface.

Modularity: A callable interface at the level of semantic interpretation will be available soon.

Extensibility: The Learner provides for easy porting to various domains. Parlance and Learner both allow for easy vocabulary extension.

Size: We recommend 16Mb memory, 50Mb disk space.

Implementation: Common Lisp.

Platform: VAX/VMS, Sun3, Sun4, SPARCstation/SunOS.

Languages: English.

Retargetability: no.

Orthography: -.

Examples: 1000-10000 sentences.

Status:

- ongoing development.
- stable.
- production quality.

Documentation:

- BBN Parlance Interface Software System Overview (available from BBN).
- Parlance User's Manual.
- Learner User's Manual.

Upgrades: provided.

SourceCode: not provided.

Consulting: available.

Support: hot line.

Format: tape.

Price:

- Parlance: $5,000 to over $100,000 depending on platform.
- Learner: $15,000 to over $100,000 depending on platform.

Restrictions: see license agreement.

Contact: Dr. Madeleine Bates.

Address: BBN Systems and Technologies / Speech and Natural Language Processing / 10 Moulton St. / Cambridge, MA 02138 / U.S.A.

Telephone: +1-617-873-3787

Email: BATES@bbn.com

PENMAN

License: research.
Authors: Dr. Eduard Hovy.

Task:

- generation.
- parsing.
- text planning.
- sentence planning.
- machine translation.

Description: The Penman project is a computational investigation into language using as theoretical bases the ideas of Systemic-Functional Linguistics, Rhetorical Structure Theory, and ideas from Artificial Intelligence on planning and knowledge representation. The project started in 1982 focusing primarily on sentence generation and English grammar building in the tradition of Systemic Linguistics. Research on text planning and parsing has been conducted since 1987. The basis for the text planning work is a merging of interclausal relations identified by Rhetorical Structure Theory (and subsequently amended), early AI work on planning, and work by Cohen and Levesque and others on representation of beliefs.

The parsing work is based on the use of classification in Loom to perform inferential functions similar to unification, including the ability to perform inference over disjunctions and negation. Much of this work is conducted in collaboration with partners at two sites in Germany and at the University of Sydney. The current main focus of the parsing and generation work is machine-aided translation, in collaboration with groups from CMU and the CRL at New Mexico State University.

Components:

- generator: generating English from semantic-like input.
- text structure planner: planning content and structure of multisentence English paragraphs.
- sentence planner: planning structure and phrasing of English sentences.

- grammar: Systemic-Functional grammar of English (Nigel) used by parser and generator.
- grammar: experimetal grammars of German, Chinese, and Japanese, in approximately same format as Nigel.
- parser: syntactic and semantic interpretation of English.
- knowledge representation: system (Loom) member of KL-ONE family.
- general high-level ontological concept taxonomy (Upper Model).
- wide-ranging semantic lexicon taxonomy (Middle Model).

- English word lexicon associated with Middle Model.
- domain model acquisition module (Uppermost).
- lexicon acquisition module (LapItUp).

Modularity:

- generator separately available (already distributed to over 80 research and university sites worldwide).
- text and sentence planners separately available.
- English grammar already distributed to numerous sites worldwide.
- Upper Model concept taxonomy available, distributed to numerous sites.
- Middle Model and extensive lexicon still under construction.
- lexicon acquisition tool separately available.
- Loom Knowledge Representation system separately available (a separate project; has been distributed to numerous sites worldwide).
- loom Knowledge Representation system separately available (it's a separate project; has been distributed to numerous sites worldwide).

Extensibility: Modules can all be extended with various amounts of difficulty.

Size: Generator, with knowledge representation system, fits on about 7 megabytes. Other components all smaller.

Implementation: Common Lisp.

Platform: Sun SPARCstations, Sun 4s, Mac-IIs, Lisp Machines. Generator, grammar, knowledge representation, Upper Model, lexicon and Domain Model acquisition tools all run on TI Explorer and Symbolics Lisp machines, Sun 3 and 4, Macintosh-II. Parser and text planner on Lisp machines only.

Languages: Other grammars (English or other languages) in the tradition of Systemic-Functional Linguistics can be adapted to fit in fairly easily; grammars of German and Japanese are currently being built in Germany and Australia by our collaborators.

Retargetability: -.

Orthography: -.

Examples:

- Generator: tested at ISI on 5 domains, over 1,000 sentence types.
- Distributed to over 80 sites worldwide and actively used by some of them.
- Parser: prototype tested on some tens of sentences at ISI; still under development.
- Text and Sentence Planners: prototypes tested on three domains at ISI; between 10 and 50 paragraphs planned and generated.
- Upper Model: contains over 200 concepts.
- Middle Model and English lexicon projected to approx. 40,000 entries by June 1993.

Status: All the projects are under further development:

- Generator: extended at ISI and at IPSI in Germany.
- Parser: extended at ISI and Ohio State University.
- Grammar: growth at ISI and Linguistics Department, University of Sydney.
- Upper Model: development at ISI and IPSI, Germany.
- Middle Model: development at ISI.
- Domain Model and lexicon tools: development at ISI.

Documentation: Numerous publications (over 70 in refereed journals, conferences, and workshops in last 3 years), including:

Generation:
- Matthiessen, C.M.I. and Bateman, J.A. Text Generation and Systemic-Functional Liunguistics. Pinter, 1991.

Text Structure Planning:
- Mann, W.C. and Thompson, S.A. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. Text Vol. 8:3, 1988. - Hovy, E.H. Planning Coherent Multisentential Text. Proceedings of the 28th ACL conference, Buffalo, 1988.

Parsing:
- Kasper, R.T. An Experimental Parser for Systemic Grammars. Proceedings of Coling 1988.

Grammar:
- Kasper, R.T. Systemic Grammar and Functional Unification Grammar. In Systemic Functional Approaches to Discourse, J. Benson and W. Greaves (eds). Ablex 1988.

Other Issues:
- Bateman, J.A. and Paris, C.L. Phrasing a Text in Terms the User can Understand. Proceedings of IJCAI 1989.

System Documentation :

- The Penman Primer (approx. 50 pp).
- The Penman User Guide (approx. 40 pp).
- The Penman Reference Manual (approx. 25 pp).
- The Nigel Manual (approx. 90 pp, short form; approx. 400 pp, long form).

Upgrades: provided.

SourceCode: provided (selected sites).

Consulting: not available.

Format: selected sites: Sun system and Lisp machine tapes, Macintosh diskettes, paper.

Price: nominal cost (selected sites).

Restrictions: Licensing agreement prohibits use for commercial purposes and protects USC/ISI from litigation and other obligations.

Contact: Dr. Eduard Hovy.

Address: Information Sciences Institute of the University of Southern California / 4676 Admiralty Way / Marina del Rey, CA 90292-6695 / U.S.A.

Telephone: +1-213-822-1511.

Email: hovy@isi.edu

PLAIN+

License: research, (partially) commercial.
Authors: Peter Hellwig, Friederike Benjes, Klaus-Georg Deck.

Task:

- linguistic analysis.
- test of linguistic theory (Dependency Unification Grammar).
- advanced computing (parallel processing, client-server architecture).
- applications (text processors, grammar checking and correction.
- information retrieval, corpus tagging, CALL-programs).

Description: PLAIN+ is a comprehensive NLP system based on Dependency Unification Grammar. It is characterized by its reliance upon lexical information, its ability to cope with difficult phenomena, e.g. coordination, its suitability for parallel processing and its capability to correct input without a previous inventory of errors. The semantic component tranforms certain natural language input into inference rules which are stored in a knowledge base and executed if necessary. The system can also be used to develop prototypes of machine translation systems and question answering systems.

The lexical resources of the system are available as a separate module. This module comprises a lemmatizer (which also decomposes compounds) and produces tagged text in SGML format. Another of its facilities is the morphology-aid function which supplies the grammatical features of any word forms as well as a generation of the whole inflectional paradigm of any word, if desired. There is also a training function which allows the user to fill in inflectional paradigms presented by the system.

Components:

- morphological analyzer/generator.
- lemmatizer including compound analysis (producing SGML tagged text).
- "pretty" hyphenator.
- parser (under development).
- semantic interpreter (under development).

data:

- lemmata of the German language with morpho-syntactic and valency information.
- lemmata of English with morho-syntactic information.
- French lexicon is in preparation.

Modularity: available independently are:

- morphological analyzer/generator.
- lemmatizer including compound analysis (producing SGML tagged text).
- "pretty" hyphenator for German.

data components are independent of program.

Extensibility: extensible by programmer.

data extensible by the linguist and any new user. There is a very convenient lexicon updating facility.

Size:

- 6 man years of work.

data:

- 13,000 most frequent lemmata of the German language.
- 6.000 most frequent lemmata of English.

Implementation: C.

Platform: UNIX (Sun), DOS (PC).

Languages: German, English, French.

Retargetability: inflectional languages with changing stems, prefixes, compounds (like German).

Orthography: fixed, 8-bit – ISO (ASCII+German).

Examples: 13.000 words with all their forms tested.

Status:

- large research.
- aiming towards production quality.
- stable.
- ongoing development.

Documentation: Peter Hellwig: Automatic Syntax Checking (March 1992).

Henriette Visser: Lexical Resources of the German Dependency Unification Grammar (March 1992).

Peter Hellwig: Pretty Hyphenation for German (October 1992).

Peter Hellwig: The Morphology-Aid Function.

User documentation in preparation.

System documentation: PLAIN+ Technical Description (confidential).

Upgrades: none.

SourceCode: available.

Consulting: available.

Format:

- tape.
- floppy disks.

Price: free of charge for research purposes.

Restrictions: all property rights reserved.

Contact: Peter Hellwig

Address: University of Heidelberg / Institute for Computational Linguistics / Karlstrasse 2 / 69117 Heidelberg / Germany

Telephone: +49-6221-543-245

Email: c87@vm.urz.uni-heidelberg.de

PROFGLOT

License: research, commercial.
Authors: Simon C. Dik and Peter Kahrel

Task:

- linguistic analysis.
- test of linguistic theory.
- machine translation.

Profglot is Prolog implementation of Simon Dik's Functional Grammar. The program is used among other things to test the theory. It generates sentences, parses sentences, translates sentences, and it performs a number of logical tasks. The program handles English, German, Danish, Dutch, French, Spanish Galician, and Japanese.

Description: The goal is to build a natural language handler, in Prolog, based on the theory of Functional Grammar described by Simon Dik. The project is useful in that it can be used to test claims and hypotheses made in the theory. Functional Grammar has "existed" since about 1978, and implementing it has led to a number of improvments to the theory.

Components:

- parser/generator.
- morphological analyzer/generator
- semantic interpreter.
- knowledge representation.
- discourse structure.
- pragmatic features.
- logic, translation.

Modularity: none.

Extensibility: extensible by developers, computational linguists and experienced users.

Size:

- 10K lines of code
- no executable code (only prolog code)

Implementation: Prolog.

Platform: MS-DOS.

Languages: Theoretically: in principle: unlimited. But ongoing research has to show the extent to which this is true. So far, the program handles a number of typologically different languages (Romance, Germanic, Altaic).Preliminary tests with typologically different languages are in progress (Turkish, Arabic).

Retargetability: Data firmly embedded in program. Data:
for each supported language, a small but semantically varied lexicon is provided: circa 30 nominal predicate frames, circa 5 adjectival predicate frames, and verbal predicate frames. The lexicon also contains idioms, meaning definitions, and irregular forms of verbal, adjectival, and nominal declensions/conjugations. Augmentation: user.

Orthography: extended IBM character set (DOS).

Examples: -.

Status:

- stable.
- ongoing development.

Documentation: Simon Dik and Peter Kahrel (1992). Profglot: a multi-lingual natural language processor. Working papers in functional grammar 45.
Simon Dik (1992). Functional Grammar in Prolog: an integrated implementation for English, French, and Dutch. Berlin & New York: Mouton de Gruyter.
Simon Dik (1992). Nederlandse Functionele Grammatica in Prolog. Amsterdam: Amsterdam Linguistic Software.

Upgrades: provided.

SourceCode: provided.

Consulting: avaiable.

Format: Prolog source code, humble additional amount (HFL 25).

Price: 750 Dutch Guilders.

Restrictions: None. Users get source code and apart from copying the program for others, are free to change the program to experiment, or to enhance the program.

Contact: Peter Kahrel.

Address: Amsterdam Linguistic Software / P.O. Box 3602 / 1001 AK Amsterdam / Netherlands

Telephone: +31-20-683-6765

Email: amling@sara.nl

Pundit

License: research.

Authors: Deborah Dahl.

Task:

- database interface.
- parsing.
- understanding.
- message processing.
- expert system interface.

Description: modularity, clean separation of domain independent and domain dependent modules.

- very large grammar.
- domain and application independent pragmatic analysis.

Components:

- parser/generator.
- semantic interpreter.
- knowledge representation.
- discourse structure.
- pragmatic features.
- temporal analysis.

Modularity: Normally we provide the whole system, but we could decouple modules if desired.

Extensibility: A knowledgeable computational linguist should be able to extend the components without too much difficulty.

Size:

- domain-independent: 60K lines of code.
- domain-dependent: about 40K lines of code.

Implementation: Prolog (currently Quintus Prolog 2.5.1).

Platform: UNIX 4.1 Sun3 or Sun4.

Languages: English.

Retargetability: Has not been tried, but it should be possible.

Orthography: -.

Examples: 1000-10000 - robust or production quality system.

Status: ongoing.

Documentation: many research papers. User documentation:

- User's guide.
- Lexical entry guide.
- research papers- can send bibliography to interested people.

Upgrades: not provided.

SourceCode: provided.

Consulting: not available.

Support: we can try to work something out on a case by case basis.

Format: tape.

Price: free, license required.

Restrictions: research or educational purposes only.

Contact: Deborah Dahl.

Address: Unisys / Center for Advanced Information Technology / PO Box 517 / Paoli PA 19301 / U.S.A.

Telephone: +1-215-648-2027

Email: dahl@prc.unisys.com

QPATR

License: research, education.

Authors: James Kilbury, Petra Barg, Ingrid Renz, Christof Rumpf.

Task:

- linguistic analysis.
- test of linguistic theory (which make use of non-monotonic inhertit.).
- learning of new lexical entries.

Description: QPATR is an implementation of the PATR-II formalism (Shieber 86) with some logical extensions. It can also be used to analyse NL sentences which include unknown words. With the use of contextual information new lexical entries for those words are constructed. The system includes an interface between PATR and DATR.

Components:

- parser.
- knowledge representation.
- DATR interpreter (non-monotonic inheritance networks).
- PATR interpreter (unification of complex feature structures).
- graphical output of constituent and feature structures.

Example grammars are provided with the system.

Modularity: the QDATR DATR Interpreter is available independently.
data components are independent of program.

Extensibility: program extensible by:

- the developer.
- the computational linguist.
- the linguist.
- the programmer.
- the experienced user.

data easily extensible.

Size:

- 230 KB source code.
- 465 KB executable.

Implementation: Prolog.

Platform: DOS.

Languages: PATR grammars.

Retargetability: no theoretical or technical limits.

Orthography: fixed, 7-bit – ISO.

Examples: -.

Status:

- demonstration.
- small research.
- stable.
- ongoing development.

Documentation: Shieber, Stuart (1986): An Introduction to Unification-Based Approaches to Grammar. CSLI Lecture Notes Vol.4, University of Chicago Press, Chicago.

Kilbury, James (1990): QPATR and Constraint Threading, COLING-90, Vol.3.

Kilbury, James / Närger, Petra / Renz, Ingrid (1992): New Lexical Entries for Unknown Words, Universität Düsseldorf.

System and user documentation: QP2MAN.TXT and *.HLP files, context-sensitive online help.

Upgrades: available.

SourceCode: none.

Consulting: available.

Format: disk.

Price: free.

Restrictions: non-commercial only, no further distribution without permission.

Contact: James Kilbury, Petra Barg, Ingrid Renz, Christof Rumpf.

Address: Heinrich-Heine-Universität Düsseldorf / Seminar für Allgemeine Sprachwissenschaft / Universitätsstrasse 1 / 40225 Düsseldorf / Germany

Telephone: +49-211-311-2557

Email: kilbury@ze8.rz.uni-duesseldorf.de, barg@ze8.rz.uni-duesseldorf.de, rumpf@ze8.rz.uni-duesseldorf.de

SCISOR / NLToolset

License: unavailable.
Authors: Lisa F. Rau.

Task:

- understanding.
- database generation (From Text).
- topic analysis / message routing.

Description: Our approach to extracting and deriving useful information from text is to use a knowledge-based, domain-independent core of text processing tools, customizing our existing programs to each new task. This core set of programs is called the NLToolset. The NLToolset derives from a research effort aimed at preserving the capabilities of natural language text processing across domains. The program achieves this transportability by using a core knowledge base and lexicon that adapts easily to new applications, along with a flexible text processing strategy tolerant of gaps in the program's knowledge base.

The design of the NLToolset combines artificial intelligence (AI) methods, especially natural language processing, knowledge representation, and information retrieval techniques, with more robust but superficial methods, such as lexical analysis and word-based text search. This approach provides the broad functionality of AI systems without sacrificing robustness or processing speed. In fact, the system has a throughput for real text greater than any other text extraction system we have seen, while providing knowledge-based capabilities such as producing answers to English questions and identifying key conceptual roles in the text (such as who did what to whom).

The NLToolset's design provides each system component with access to a rich hand-coded knowledge base, but each component applies the knowledge selectively, avoiding the computation that a complete analysis of each text would require. The architecture of the system allows for levels of language analysis, from rough skimming to in-depth conceptual interpretation.

Components:

- morphological analyzer/generator.
- parser/generator.
- semantic interpreter.
- knowledge representation.
- lexical analyzer.
- conceptual retrieval mechanism.

Modularity: not available.

Extensibility: easily extensible by developers.

Size: 50,000+ lines of code, 20,000 knowledge base entries.

Implementation: Lisp.

Platform: Sun/UNIX.

Languages: English.

Retargetability: not easily.

Orthography: -.

Examples: 1000-10000.

Status:

- completed.
- ongoing development in terms of tailoring to new domains.
- production quality.

Documentation: P. S. Jacobs and L. F. Rau. Scisor: A system for extracting information from on-line news. Communications of the Association for Computing Machinery (ACM), forthcoming, 1990.

P. S. Jacobs and L. F. Rau. The GE NLToolset: A Software Foundation for Intelligent Text Processing. In Proceedings of the Thirteenth International Conference on Computational Linguistics, Helsinki, Finland, 1990.

L. F. Rau. Conceptual information extraction and retrieval from natural language input. In Proceedings of the User-Oriented Content-based Text and Image Handling (RIAO) Conference, Boston, MA, March, 1988.

NLToolset manual (GE proprietary).

Upgrades: provided.

SourceCode: provided.

Consulting: available.

Format: -.

Price: -.

Restrictions: This software is unavailable to the public. All support information pertains to GE internal use.

Contact: Lisa F. Rau.

Address: General Electric Corporate Research and Development / Artificial Intelligence Program / P.O. Box 8 / 1 River Road / Schenectady, NY 12301 / U.S.A.

Telephone: +1-518-387-5059

Email: rau@crd.ge.com

SNePS

License: research.
Authors: Stuart C. Shapiro, William J. Rapaport, Jeannette G. Neal.

Task:

- machine translation.
- database interface.
- parsing.
- generation.
- understanding.
- planning.
- acting.

Description: (1) The SNePS Semantic Network Processing System, a knowledge- representation/reasoning system that allows one to design, implement, and use specific knowledge representation constructs, and which easily supports nested beliefs, meta-knowledge, and meta-reasoning.

(2) SNIP, the SNePS Inference Package, which interprets rules represented in SNePS, performing bi-directional inference, a mixture of forward chaining and backward chaining which focuses its attention on the topic at hand. SNIP can make use of universal, existential, and numerical quantifiers, and a specially-designed set of propositional connectives that include both true negation and negation-by-failure.

(3) Path-Based Inference, a very general method of defining inheritance rules by specifying that the existence of an arc in a SNePS network may be inferred from the existence of a path of arcs specified by a sentence of a "path language" defined by a regular grammar. Path-based reasoning is fully integrated into SNIP.

(4) SNeBR, the SNePS Belief Revision system, based on SWM, the only extant, worked-out logic of assumption-based belief revision.

(5) A Generalized Augmented Transition Network interpreter/compile that allows the specification and use of a combined parsing- generation grammar, which can be used to parse a natural-language sentence into a SNePS network, generate a natural-language sentence from a SNePS network, and perform any needed reasoning along the way.

(6) A theory of Fully Intentional Knowledge Representation, according to which we are developing knowledge representation constructs and grammars for the Computational Cognitive Mind. This theory also affects the development of successive versions of SNePS and SNIP. For instance, the insight we developed into the intentional nature of rule variables led us to design a restricted form of unification that cuts down on the search space generated by SNIP during reasoning.

(7) CASSIE, the Computational Cognitive Mind we are developing and experimenting with, successive versions of which represent an integration of all our current work.

Current research projects include:

- VMES, the Versatile Maintenance Expert System.
- discussing and using plans (expressed in sentences).
- intelligent multi-media interfaces.
- cognitive and computer systems for understanding narrative text.
- the representation of natural category systems and their role in natural-language processing.
- belief representation, discourse analysis, and reference in narrative.
- understanding pictures with captions.
- automatic acquisition of word meanings from NL contexts
- Issues of Semantics in a Semantic Network Representation of Belief
- Knowledge Representation with Structured Variables
- Belief Ascription by Way of Simulative Reasoning
- Representations of and Reasoning about Collections
- Experience-Based Learning in Deductive Reasoning Systems
- Interactive Generation of Plan Descriptions and Justifications
- Representing and Learning Successful Routine Activities
- From Beliefs and Goals to Intentions and Actions

Components:

- morphological analyzer/generator.
- parser/generator.
- semantic interpreter.
- knowledge representation.
- discourse structure.
- pragmatic features.
- reasoning.
- belief revision.

Modularity: firmly embedded.

Extensibility: extensions constitute research issues.

Size: -.

Implementation: Common Lisp.

Platform: UNIX.

Languages: English.

Retargetability: -.

Orthography: -.

Examples: -.

Status: ongoing.

Documentation: A bibliography of over 100 published articles, techn- ical reports, and technical notes may be obtained from Prof. Shapiro or Prof. Rapaport, at the address given below, or by electronic mail to shapiro@cs.buffalo.edu or rapaport@cs.buffalo.edu

Upgrades: not provided.

SourceCode: not provided.

Consulting: not available.

Support: -.

Format: -.

Price: -.

Restrictions: -.

Contact: Stuart C. Shapiro.

Address: SUNY Buffalo / Dept. of Computer Science / 226 Bell Hall / Buffalo, NY 14260 / U.S.A.

Telephone: +1-716-636-3180

Email: shapiro@cs.buffalo.edu, rapaport@cs.buffalo.edu

SUNDIAL

License: research.
Authors: G. Niedermair.

Task:

- linguistic analysis.
- database interface.

Description: Linguistic Processor for spoken Language Applications.

Components:

- phonological analyzer.
- knowledge representation.
- discourse structure.
- parser.
- semantic interpreter.

data:

- German lexicon.
- German grammar.

Modularity: speech analyser, linguistic ananlyser and dialogue module are available independently.

data components independent of program.

Extensibility: program and data extensible by:

- the developer.
- the computational linguist.
- the programmer.

Size: data:

- 1000 lexicon entries.
- 300 TUG-grammar rules.
- 100 types for semantic interpretation in type hierarchy.
- 100 rules for semantic interpretation.

Implementation: Prolog.

Platform: UNIX/Sun.

Languages: German.

Retargetability: -.

Orthography: -.

Examples: about 1000 sentences tested.

Status:

- demonstration.
- stable.
- ongoing development.

Documentation: scientific publications.

Upgrades: none.

SourceCode: none.

Consulting: none

Format: -.

Price: -.

Restrictions: limited to project members .

Contact: G. Niedermair

Address: Siemens AG / ZFE ST SN 74 / Otto-Hahn-Ring 6 / 81739 München / Germany

Telephone: +49-89-6362-374

Email: nie@zeisig.zfe.siemens.de

YAKR

License: commercial.

Authors: Lutz Prechelt, Rolf Adams and Finn Dag Buoe.

Task:

- taxonomic knowledge representation.
- query answering.

Description: Very simple NL knowledge aquisition, powerful domain knowledge representation.

Components:

- parser (case frame parser).
- morphological analyzer (full form dictionary).
- semantic case frame interpreter
- knowledge representation.

Modularity: -.

Extensibility: can be extended easily.

Size: 45K lines of code

Implementation: C++.

Platform: SUN-OS, portable.

Languages: German.

Retargetability: Data partially embedded in program, partially independent of program.

Orthography: -.

Examples: -.

Status:

- almost stable.
- no continuing development.

Documentation: Wissensrepräsentation und -akquisition in einem natürlichsprachlichen Softwareinformationssystem, Rolf Adams, Institut für Programmstrukturen und Datenorganisation, Universität Karlsruhe, 1992

The SIS Project: Software Reuse with a Natural Language Approach, Lutz Prechelt, Institut für Programmstrukturen und Datenorganisation, Universität Karlsruhe, 1992

Software Engineering and its Applications, Engineering Cost-Effective Natural Language Interfaces for Knowledge Based Systems, Lutz Prechelt and Finn Dag Buo and Rolf Adams, 1992

Conference on Artificial Intelligence Applications, Transportable Natural Language Interfaces for Taxonomic Knowledge Representation Systems, Lutz Prechelt and Finn Dag Buo and Rolf Adams, IEEE, 1993

Upgrades: none.

SourceCode: available.

Consulting: none.

Format: unix tar.

Price: free.

Restrictions: GNU general public licence.

Contact: Lutz Prechelt.

Address: Universität Karlsruhe / Institut für Programmstrukturen und Datenorganisation/ Am Fasanengarten 5/ 76131 Karlsruhe / Germany

Telephone: +49-721-608-4317

Email: prechelt@ira.uka.de

11 NLP-Tools

COMPULEXIS

License: commercial, research.
Authors: H. Madsen, E. Vennebush.

Task: lexicography, dictionary text processing.

Description: Capturing and processing of dictionary data, using tags for the different data types. On line editing and search facilities. Output to printer or as photocomposition file. IPA font available.

Components:

- text processor.
- text analyzer.

Modularity: firmly embedded.

Extensibility: -.

Size: -.

Implementation: C, Oracle.

Platform: MS-DOS

Languages: Any European.

Retargetability: -.

Orthography: -.

Examples: 1000 - 10000 dictionaries.

Status:

- existing version completed.
- ongoing new releases; New version under development.
- stable.
- production quality.

Documentation: Descriptive Tools for Electronic Processing of Dictionary Data, Vol 20, Lexicographica Series Maior, 1987, Max Niemeyer Verlag Germany.
Literary and Linguistic Computing Vol 3, no 3 1988. Oxford University Press.

Upgrades: provided.

SourceCode: not provided.

Consulting: available.

Support: telephone support.

Format: 3.5 or 5.25 inch disks.

Price: from 5000 pounds.

Restrictions: has to be configured specifically for each project. Price varies accordingly.

Contact: H. Madsen, E. Vennebush.

Address: Moor's Edge / Charlton-on-Otmoor / Oxford OX5 2UG / UK

Telephone: +44-867-33550

Email: -.

CUF

License: research.

Authors: Michael Dorna, Jochen Dörre.

Task: CUF provides a formalism for declarative description of linguistic phenomena independent of the linguistic area and a system for processing CUF descriptions.

Description: declarative description of linguistic phenomena (syntax, phonology, morphology and semantics) in the CUF formalism and constraint-based processing of linguistic knowledge in the CUF system.

Components:

- compiler.
- interpreter/ constraint solver.

Data:

- HPSG-based grammar (about 150 CUF clauses).
- GB-based mini-grammar (about 50 CUF clauses).

Modularity: Data components are independent of the program.

Extensibility: The system can be extended by:

- the developer.
- the computational linguist.
- the programmer.

Data components can be extended by:

- the computational linguist.
- the linguist.
- the experienced user.

Size: 200KB executable.

Implementation: Quintus Prolog.

Platform: UNIX.

Languages: German.

Retargetability: applicable for all natural languages.

Orthography: ASCII.

Examples: about 10^3 words/phrases/sentences tested.

Status:

- small research.
- demonstration.
- stable.
- continuing development.

Documentation:

- Diploma Thesis and small paper in German.
- report and short documentation in English.

Upgrades: none.

SourceCode: none.

Consulting: none.

Format: compressed tar file (by anonymous ftp from ftp.ims.uni-stuttgart.de).

Price: free.

Restrictions: non-profit.

Contact: Michael Dorna, Jochen Dörre.

Address: Universität Stuttgart/ Institut für maschinelle Sprachverarbeitung (IMS)/ Azenbergstrasse 12/ 70174 Stuttgart / Germany

Telephone: -.

Email: michel@ims.uni-stuttgart.de, jochen@ims.uni-stuttgart.de

DCG workbench

License: research.

Authors: Peter Klima, Konrad Jablonski.

Task:

- linguistic engineering.
- comfortable development of DCG grammar rules.

Description: The DCG workbench compiler was constructed to support the grammar rule development for the product NUGGET (text generation system). The linguistic engineer can use a high level grammar language similar to parts of LFG and GPSG and can code grammar rule frames. These frames are automatically compiled to DCG rules. The semantic-pragmatic restrictions are inserted as described by the LE. The calls of the morphological component are inserted automatically for each terminal rule. Global checks and consistency tests are done.

The modular construction of the system allows enhancement of the notation for the high level grammar formalism in an easy way by changing an EBNF notation. So the DCG workbench supports not only extension of the language specific grammar rules but also the modification / extension of the high level description language.

The system was designed for the transfer of grammars to different languages and was tested by the transfer of NUGGET grammar to the Italian.

Components: compiler (high level grammar language to augmented DCG).

Modularity: it is one component.

Extensibility: extensible by any compiler developer.

data can be extended by a (computational) linguist.

Size: used systems for compiler construction: Parser-Generator (Meta-Checker, Parser-Tables-Assembler, LL(1)-Tables-Generator):

- 14000 lines of source code.
- 200 KB of executable.

on top of this: DCG workbench (= compiler: high level grammar to augmented DCG rules with sem./prag. restrictions):

- 20000 lines of source code.
- 180 KB of executable.
- 1,5 man years of work.

Implementation: C & Pascal.

Platform: TARGON OS (UNIX) .

Languages: -.

Retargetability: -.

Orthography: fixed, 8-bit – proprietary ASCII.

Examples: 500 different grammar rules in german & italian were processed.

Status:

- demonstration.
- large research.
- stable.
- no continuing development.

Documentation: Peter Klima: internal project documents (EG deliverables).

K.Jablonski, J.Samuel: Natural Language Information Access System; in: ESPRIT - Information Processing Systems; Brussels 1990.

Upgrades: none.

SourceCode: none.

Consulting: available.

Support: application projects.

Format: on demand.

Price: individual project rates.

Restrictions: only for the modification of the workbench (not of the grammar rules): Pyramid Pascal.

Contact: Konrad Jablonski.

Address: Sietec Consulting GmbH / Business unit of Siemens Nixdorf Informationssysteme AG / Projekt-Zentrum Expertensysteme & KI / Riemekestr. 160 / Postfach 21 60 / 33106 Paderborn / Germany

Telephone: +49-52518-31863, Fax: -31869

Email: jablonski.pad@sni.de (EUNET)

Dictionary Maintenance Utilities

License: commercial.
Authors: Ken Litkowski.

Task: dictionary maintenance for any NL system requiring a dictionary.

Description: The utilities provide a flexible tool for creating and maintaining dictionaries, so that student or researcher does not need to spend time reinventing dictionary formats. In addition, user can easily experiment with different theoretical constructs for dictionary organization. (Dictionaries have been used with unification-based grammars and have supported complex lexical hierarchies, such as used in HPSG.)

Components: Can accept most dictionary information used in NL processing, including ordinary dictionary data, features, roles, hierarchical links (super-, sub-, multiple), and semantic interpretation rules, with multiple senses allowed for each entry.

Processing components allow for (1) adding, deleting, or viewing dictionary entries, (2) listing the entries, (3) converting entries into LISP, Prolog, or ASCII format, (4) testing a word's features and selectional restrictions (following hierarchical paths), and (5) merging multiple files (e.g., several student-created files).

Modularity: Dictionary data structure is firmly defined. Processing components are modularized.

Extensibility: Program is written in C and source code is provided in well-defined and documented modules, allowing for easy modification by anyone with a C compiler.

Size:

- source code
- 6,000 lines.
- executable programs
- up to 50K.

Implementation: C.

Platform: MS-DOS.

Languages: English.

Retargetability: not tested, but should be OK.

Orthography: -.

Examples: Dictionaries up to 500 entries have been built, but system is sufficiently robust to handle hundreds of thousands of entries, using B-tree access methods.

Status: Ongoing. Version 2 will be available in 2nd quarter of 1991, with interface to machine-readable dictionary (Merriam-Webster's Concise Electronic Dictionary).

Documentation: Users manual is included. System is described in detail through the descriptions of each function included in the source code.

Upgrades: provided.

SourceCode: provided.

Consulting: available.

Support: Program has been tested over two semesters in graduate-level NLP course. Most bugs have been fixed. Source code is provided. Customization and consulting support are available. Upgrades are planned, with a significant extension due in 2nd quarter of 1991 which will allow processing of a stream of text against user-generated and machine-readable dictionaries.

Format: PC disks (size should be specified by customer).

Price:

- US$50.00 for version 1 for individual use.
- US$250.00 for classroom use, with each student allowed to become registered owner.
- US$125.00 (tentative) for version 2, with discount available to registered owners of version 1.
- negotiable for business use.

Restrictions: No restriction on dictionaries converted to Lisp, Prolog, or ASCII Individual can make any desired modifications to source, but cannot give or market source or modifications to others (license agreements are available).

Contact: Ken Litkowski.

Address: CL Research / 20239 Lea Pond Place / Gaithersburg, MD 20879/ U.S.A.

Telephone: +1-202-275-1553 or +1-301-926-5904

Email: 71520.307@compuserve.com

Dictionary Maintenance Programs

License: research, commercial.
Authors: Ken Litkowski.

Task: evaluation of possible antecedents for anaphoric pronouns in texts, developed as a component of an experimental machine translation system.

Description: DIMAP is intended to capture the state-of-the art in the development of computational lexicons. It is based on the theory that the lexicon can contain almost all data necessary for natural language processing (as in HPSG). It is also based on the theory that the lexicon is highly structured into an inheritance hierarchy. The software is is ideal for student use in designing lexicons for natural language processing, particularly in combination with the users manual which provides an introduction to principles of computational lexicology.

Components: program:

- 1) Merriam-Webster Concise Electronic machine-readable dictionary consisting of 80,000 entry points for 22,000 lemmas, available for (1) tagging text with part-of-speech and some features and (2) selective or batch conversion to machine-tractable DIMAP format.
- 2) Ability to add new entries with multiple senses in DIMAP format, each containing dictionary lexicographic information, superclass and instance links, feature and role lists, and semantic interpretation rules (using Allen's logical forms).
- 3) Tag text with information (part-of-speech, inflections, features) available in combination of DIMAP and machine-readable components.
- 4) Ability to convert DIMAP dictionary information to your format (Lisp, Prolog, or user-specified ASCII format) or to upload your dictionaries to DIMAP format for easier maintenance.
- 5) Test a word's features and selectional restrictions through an inheritance hierarchy.
- 6) Ability to merge several (student) dictionaries.

data:

- 1) Machine-readable dictionary.
- 2) DIMAP-created dictionary.

Modularity: available as independent modules:

- Conversion.
- uploading.
- merging components.

available as independent data:
DIMAP-created dictionary is independent of program, although format (described in C data structures) is heavily dependent on specific C code.

Extensibility: Programs themselves cannot be extended, but C object code is included (in DOS library or UNIX archive) for incorporation by experienced programmer into programs that can extend the functionality. Object code includes routines for accessing information in DIMAP portion of the dictionary, but not for the machine-readable dictionary (because of license restrictions).

Data components can be extended by any type of user, including lexicographers, linguists, computational linguists, knowledge engineers, translators, and those involved in information retrieval.

Size: Executable programs contain about 400K bytes of code.
data:

- Machine-readable dictionary: 80,000 entry points and 22,000 lemmas.
- Four sample dictionaries.

Implementation: C.

Platform: DOS, Sun3, Sun4, ULTRIX.

Languages: English.

Retargetability: see continuing development.

Orthography: US ASCII (although simple modifications can be made upon request to accept other character sets).

Examples: Any English language ASCII text in sentence or paragraph format can be processed. Machine-readable dictionary has been shown to cover about 92.5 percent of wire service text, increasing to 99.5 percent when proper nouns are excluded. The software operates in production quality and high volume, but is also quite suitable for further research use.

Status:

- small research.
- stable.
- continuing development (incorporation of a full English parser, incorporation of WordNet data, delivery of Windows version and development of language-independent version based on Unicode).

Documentation: DIMAP is currently under review for the journals "Computational Linguistics" and "Computers and the Humanities". (Please check with CL Research for updates.)

Users manual is 130 pages in length and includes a 60 page introduction to computational lexicology, covering basic lexicons for natural language processing (Allen), principles of computational lexicography (including Mel'cuk's explanatory and combinatory dictionary, Jackendoff's lexical conceptual structures, Levin's lexical subordination, and Pustejovsky's qualia structures), type hierarchies (includeing Flickinger's lexical hierarchies for HPSG, Nirenburg's ontologies, Miller's WordNet, and Sowa's semantic networks), Schank's conceptual dependency structures, and feature structures for unification grammars. Sample dictionaries are included in the software to illustrate how the structures inherent in these theories would be implemented. Functions included in object library or archive are fully described as to function and arguments. A substantial bibligraphy and an index are included in the manual.

Upgrades: All bugs are fully dealt with and upgrades resulting are provided free of charge to those reporting any problems.

SourceCode: available for license or development purposes upon negotiation.

Consulting: available to extend DIMAP in ways that may be desired by the user.

Format:

- any size diskette for DOS or UNIX versions.
- Demonstration versions are available for DOS or Sun4 from the Consortium for Lexical Research by anonymous ftp (lexical@nmsu.edu).

Price:

- $350 for single-user copy (available only in DOS).
- $1,000 for academic or institutional copies (available in DOS or UNIX versions.) Any person in the school or institution is allowed to make a copy for individual use and hence become a single-user owner, eligible for discounts on future products.
- $2,400 for commercial copy (either DOS or UNIX versions) for single machines.
- $15 for demonstration copy directly from CL Research.
- $15 for overseas postage.

Please specify disk size. Payment by check drawn in US dollars on a US bank. Purchase orders will be accepted on official academic, institution, or business stationary. License agreements available for business use. Money back guarantee, less postage, within 45 days.

Restrictions: Copies are restricted to a single machine. Dictionaries created with these utilities are owned by the user and may be freely distributed as long as utilities are not included.

Contact: Ken Litkowski.

Address: CL Research/ 20239 Lea Pond Place/ Gaithersburg, Maryland 20879/ U.S.A.

Telephone: +1-301-926-5904

Email: 71520.307@compuserve.com

DITO – DIagnostic TOol for german syntax

License: research.

Authors: John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein.

Task:

- diagnosis of errors in the syntactic component of NLP systems.
- monitoring of performance.
- maintenance of consistency of syntactic processing.
- support of grammar development.

Description: DiTo is an ongoing effort to construct a catalogue of syntactic data exemplifying the major syntactic patterns of German. The purpose of the corpus is to support the diagnosis of errors in the syntactic components of natural language processing (NLP) systems. Secondary aims are the evaluation of NLP syntax components and support of theoretical and empirical work on German syntax.

The data consist of artificially and systematically constructed expressions, including also negative (ungrammatical) examples. The data are organized into a relational database and annotated with some basic information about the phenomena illustrated and the internal structure of the sample sentences. The organization of the data supports selected systematic testing of specific areas of syntax, faciliates the addition of data of syntactic areas, but also serves the purpose of a linguistic database.

We invite other research groups to participate in our effort, so that the diagnostics tool can eventually become public domain. Several groups have already accepted this invitation, and progress is being made.

Components:

- parser.
- generator.

data: relational database.

Modularity: Data-Files and Database Software are available independently. AQL, a high-level query language for DiTo, is firmly embedded in the program.

Extensibility: the developer: can add new data and extend the structure of the database.

the computational linguist: can work on syntactic areas according the DiTo classification.
the linguist: can work on syntactic areas according the DiTo classification.

the experienced user: can add new data.

the programmer: can modify the DiTo-Database Management System.

Size: 7 MB executable and data.

Implementation: AWK, YACC.

Platform: UNIX on SPARCstation, SunOS Release 4.1.

Languages: German.

Retargetability: none.

Orthography: ASCII.

Examples: german words and sentences tested.

Status:

- small research.
- stable.
- ongoing development.

Documentation: John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein: A Diagnostic Tool for German Syntax. DFKI Research Report RR-91-18. Saarbrücken, 1991.

User documentation:

Judith Klein und Ludwig Dickmann. DiTo-Datenbank. Datendokumentation zu Verbrektion und Koordination. DFKI-Document D-92-04, Saarbrücken 1992.

Brigitte Krenn und Martin Volk. DiTo-Datenbank. Datendokumentation zu Funktionsverbgefügen und Relativsätzen. To appear as DFKI-Document, Saarbrücken 1993.

Ludwig Dickmann und Judith Klein: DiTo-Handbuch. To appear as DFKI Technical Memo, Saarbrücken 1993.

System documentation:

Abdel Kader Diagne:DiTo-DMS.The DiTo Database Management System. Concepts, Implementation Issues and User Guide. To appear as DFKI-Report, Saarbrücken, 1993.

Upgrades: available.

SourceCode: available.

Consulting: available.

Format: ftp.

Price: cooperation for development of coverage of further syntactic areas.

Restrictions: The present test application in Machine Translation Systems is at the IAI Saarbrücken. At the moment we share the database with all interested groups that provide us data of further syntactic areas. Eventually, data will be given to all interested research groups without restrictions.

Contact: Judith Klein.

Address: German Research Center for Artificial Intelligence (DFKI) / Computational Linguistics / Stuhlsatzenhausweg 3 / 66123 Saarbrücken / Germany

Telephone: +49-681-302-5309

Email: klein@dfki.uni-sb.de

EGG – editor for GPS-grammars

License: research.

Authors: Erich Ziegler, Tim Eckhart and Wilhelm Weisweber.

Task: tool for defining linguistic data needed for syntactic analysis or generation with GPSGs.

Description: A GPS grammar can be easily defined with EGG. Consistency checks are performed and the FCRs are applied to the other components. The output can be directly used by a parser or a generator.

Components: editor.

GPSG with 5 components (represented as first order terms):

- feature definition.
- aliases.
- ID rules.
- LP statements.
- FCRs.

Extensibility: program extensible by the developer.

any computational linguist or linguist familiar with GPSG can define data.

Size:

- 2 MB executable.
- 2 man years of work.

Implementation: Arity Prolog.

Platform: MS-DOS 3.31, AT compatible PC.

Languages: a German and an English grammar developed.

Retargetability: theoretically, any language can be described in GPSG.

Orthography: -.

Examples: German and English grammar.

Status:

- not stable.
- no continuing development.
- demonstration.
- small research.

Documentation: Ch. Hauenschild, Stephan Busemann "A constructive Version of GPSG for Machine Translation" in: E. Steiner, P. Schmidt, C. Zellinsky-Wibbelt (eds.) "From Syntax to Semantics - Insights from Machine Translation", Frances Pinter, London 1988, p. 216-238.

User and system documentation in progress.

Upgrades: none.

SourceCode: none.

Consulting: available.

Format: 5 1/4" or 3 1/2" diskettes.

Price: free.

Restrictions: no.

Contact: Erich Ziegler.

Address: Technical University of Berlin / Department for Software and Theoretical Computer Sciences / KIT / Sekr. FR 5-12/ Franklinstr. 28/29 / 10587 Berlin / Germany

Telephone: +49-30-314-73604 / -27778

Email: ez@cs.tu-berlin.de

GTU – Grammatik-Test-Umgebung

License: research.

Authors: Martin Volk, Hanno Ridder, Johannes Hubrich.

Task:

- linguistic analysis.
- test of linguistic theory.

Description: GTU is an easy to use tool for teaching syntax analysis. It facilitates grammar development through many help functions, build-in morphology component and lexicon as well as an automatic output procedure. It is currently set up for DCG-style, ID/LP-style grammars with feature structures as well as LFG.

Components:

- morphological analyzer.
- parser.

Modularity: no.

Extensibility: can be extended the developer.

Size: 400 kilobytes of executable.

Implementation: Prolog.

Platform: MS-DOS 3.0 or greater.

Languages: German.

Retargetability: -.

Orthography: -.

Examples: 100 examples processed successfully (words and sentences).

Status: -.

Documentation: Martin Volk ,The role of testing in grammar engineering, Proceedings of the third conference on applied natural language processing, Trento 1992, 257-58, 1992.

Martin Volk und Hanno Ridder, GTU - Eine Grammatik Testumgebung mit Testsatzarchiv, LDV-FORUM 9(1992), 1:34-37, 1992.

Upgrades: available.

SourceCode: available.

Consulting: not available.

Format: 5.25 inch or 3.5 inch floppy disk.

Price: 20,- DM.

Restrictions: none.

Contact: Martin Volk.

Address: University of Koblenz-Landau / Computational Linguistics / Rheinau 3-4 / 56075 Koblenz / Germany

Telephone: +49-261-9119-469

Email: volk@infko.uni-koblenz.de

GULP – Graph Unification Logic Programming

License: research.

Authors: Michael A. Covington.

Task: grammar implementation tool.

Description: It solves a pesky problem with Prolog, i.e., the lack of a good way to represent feature structures in which features are identified by name rather than position.

Components: A simple extension to Prolog giving a convenient representation of feature structures (attribute-value structures). This can be used with Prolog's built-in top-down parser or with any parser written in Prolog.

Modularity: -.

Extensibility: by programmer.

Size: 1172 lines of source code.

Implementation: Prolog (Quintus, Arity, ALS).

Platform: Any with Edinburgh-compatible Prolog.

Languages: all.

Retargetability: see above.

Orthography: -. Data: none except a few very short example grammars.

Examples: (system is production quality / high volume).

Status: stable and continuing.

Documentation: Research Report AI-1989-01, Artificial Intelligence Programs, The University of Georgia, Athens, Georgia 30602.

Upgrades: not provided.

SourceCode: available.

Consulting: not available.

Format: Anonymous ftp from aisun1.ai.uga.edu, directory ai.reports.

Price: none.

Restrictions: To be freely shared among scholars. Not for commercial resale.

Contact: Michael A. Covington.

Address: Artificial Intelligence Programs / Graduate Studies Research Center / The University of Georgia / Athens, Georgia 30602 / U.S.A.

Telephone: +1-404-542-0359

Email: mcovingt@uga.cc.uga.edu

Linguistic DataBase

License: commercial.

Authors: Hans van Halteren

Task: database system for (syntactic analysis) tree structures.

Description: Main goal: corpus linguistic research on the syntactic level. The system enables linguists to browse through analysis trees, search for syntactic patterns, extract examples and create frequency counts at the syntactic (as compared to lexical) level. The analysis trees in the system are prestored nad verified. No heavy-duty parsing system needs to be run during exploitation.

The database system is also of use to developers of parsers by enabling them to examine the output of their products in an accessible format.

Components:

- Tree viewer for non-graphics terminals/screens.
- query language editor.
- query language processor.

Modularity: firmly embedded.

Extensibility: no.

Size:

- MS-DOS, main program binary: 200K.
- VAX/VMS, main program binary: 225K.
- UNIX, main program binary: (SUN) 400K.
- UNIX, main program C: 706K.
- packaged data: 8M.

Implementation: CDL2, C.

Platform: MS-DOS, VAX/VMS, UNIX.

Languages: English.

Retargetability:

- Western European scripts: easily.
- other: possibly.

Orthography: -.

Examples: 1000-10000 tree structures.

Status:

- completed.
- stable.
- production quality.

Documentation: van Halteren, Hans and Nelleke Oostdijk. "Using an Analyzed Corpus as a Linguistic Database", in Computers in Literary and Linguistic Computing, Proceedings of the XIIIth ALLC Conference (Norwich 1986), John Roper (vol. ed.), J. Hamesse and A. Zampolli (series eds.).

van Halteren, Hans and Theo van den Heuvel. "Linguistic Exploitation of Syntactic Databases". (Rodopi, Amsterdam 1990).

de Haan, Pieter. "Exploring the Linguistic Database: Noun Phrase Complexity and Language Variation", in Corpus Linguistics and Beyond, Willem Meijs, ed. (Rodopi, Amsterdam 1987).

Upgrades: not provided.

SourceCode: not provided.

Consulting: available.

Format:

- MS-DOS: packed files on 3.5" diskettes, binaries only.
- VAX/VMS: VAX backup format on tape or TK50, binaries only.
- UNIX: tar format, medium to be agreed upon, C files.

Price:

- academic: cost of porting (or nominal charge of Hfl. 100).
- non-academic: Hfl. 5000.

Restrictions: none.

Contact: Hans van Halteren.

Address: University of Nijmegen / Dept. of Language and Speech / Dept. of Language and Speech / 6500 HD Nijmegen / Netherlands

Telephone: +31-80-612836

Email: cor_hvh@kunrc1.urc.kun.nl

LINGUIST

License: research.
Authors: Hiroshi Sano.

Task: a Japanese linguistic development environment.

Description: LINGUIST is an interactive Japanese language analysis system on CESP which is available under UNIX. The LINGUIST computational tool was designed to assist the development of grammar rules written in DCG. LINGUIST can be used to develop grammar rules written in DCG and to modify them for use in many sort of application systems. Grammar rules can be used and modified in existing grammar for Japanese provided from ICOT's 6th Research Laboratory.

LINGUIST is interactive; you can instantly see the results of analysis on the grammar rules or dictionaries on which you made changes. LINGUIST offers many features to make analysis experiments easy and comprehensive. These features include the following:

- Minimum key entries. LINGUIST is designed to allow most operations to be performed with a mouse.
- Multiple windows. Provides high visibility flow of executions to be viewed by locating all necessary functions in one window.
- Powerful utilities. A visual debugger traces grammar rules.
- Supplied linguistic knowledge source. Basic grammars that contain grammar rules for Japanese sentence analysis and dictionaries.

LINGUIST was originally designed for use on PSI machines. Now the software is available on Sun workstations under Unix using CESP.

Components: grammar writers workbench.

data: see FJGH.

Modularity: data components are independent of program.

Extensibility: extensible by the developer and the programmer.

data extensible by:

- the developer.
- the computational linguist.
- the experienced user.

Size: 500 KB executable.

data: see FJGH.

Implementation:

- ESP (Extended Self-contained Prolog developed by ICOT).
- CESP (Common ESP developed by AI language research Corp., JAPAN).

Platform:

- (1) ESP version runs on PSI-II.
- (2) CESP version runs on Sun work-station.

Languages: Japanese.

Retargetability: none.

Orthography: 16-bit EUC code (Including KANJI code).

Examples: sentences.

Status:

- large research.
- stable.
- ongoing development.

Documentation: User documentation: LINGUIST / User's Manual (In English).

System documentation: LINGUIST / Software Manual (In Japanese).

Upgrades: none.

SourceCode: available.

Consulting: available.

Format: -.

Price: free.

Restrictions: none.

Contact: Hiroshi Sano.

Address: Kansai Research Laboratory/ Toshiba Corp. / Information Processing Gourp / Minami-machi/ Motoyama / Higaishinada-ku / 658 Kobe city / Japan

Telephone: +81-078-435-3551

Email: sano@krl.toshiba.co.jp

P–TRA

License: research.

Authors: Dr. D. Stock.

Task: phonetic transcription.

Description: The system transforms unlimited (German) text into phonetic transcription by means of an interpreter and an independent set of context-sensitive rules written in form of Boolean expressions.

Components: interpreter.

data: package of rules.

Modularity: it is one module.

data components are independent of program.

Extensibility: program extensible by the experienced or new user.

data extensible by the new user.

Size: 35 KB of executable.

data:

- set of ca. 1000 rules formulated as Boolean expressions for context-sensitive phonetic transcription of German.
- 20 KB rules.

Implementation: Fortran 77.

Platform: PC-DOS 5.0.

Languages: German.

Retargetability: no limits, rules must be provided by user.

Orthography: -.

Examples: 100000 words tested.

Status:

- large research.
- stable.
- ongoing development.

Documentation: System and user documentation: D. Stock, P-TRA - eine Programmiersprache zur phonetischen Transkription, In: W. Hess & W.F. Sendlmeier, Beiträge zur angewandten und experimentellen Phonetik, Stuttgart, 1992, p. 222 - 231.

Upgrades: available.

SourceCode: none.

Consulting: none.

Format: EXE-file with interpreter, ASCII-file with rules.

Price: 2000 DM + tax.

Restrictions: material is copyrighted.

Contact: Dr. D. Stock.

Address: Institut für Kommunikationsforschung und Phonetik / Universität Bonn / Poppelsdorfer Allee 47 / 53115 Bonn / Germany

Telephone: +49-228-735-638

Email: wgh@wgh.ikp.uni-bonn.de

SEMBLEX

License: research.

Authors: Frank Wegmann.

Task: (semantic) linguistic analysis.

Description: SEMBLEX is intended as a tool for the acquisition of lexical entries. Due to its integration into the HyperCard environment it is easy to learn and to use even for beginners. By establishing a custom structure of the lexicon it was possible to integrate different types of dictionaries into a coherent set of lexical entries. All entries thus share the same set of data items, but their structure is at the same time flexible enough to allow structural changes. Its primary goal is to support the process of semantic analysis of lexical entries. Although not having a true representation of the entries underneath, SEMBLEX offers some methods that assist the lexicographer (or the linguist) in his work. There are facilities that allow the data to be exported to other platforms. Furthermore, SEMBLEX has now been localised w.r.t. to the five languages mentioned above. It is not expected to become a lexicon with thousands of entries but should be regarded as a user-friendly construction site for semantic analysis within a project limited research framework.

Components:

- pragmatic features.
- semantic interpreter.
- acquisition component for lexicography.

lexical data for German, French and English.

Modularity: none.

Extensibility: easily extensible.

Size:

- 5-6 MB overall.
- 3000 lines of source code.
- 600 KB of executable.
- 0.8 man years of work.

data:

- 1200 entries for German.
- 800 entries for English.
- 800 entries for French.

Implementation: Hypertalk.

Platform: MacOS, MacPlus or greater.

Languages: German, English, French.

Retargetability: Spanish, Italian.

Orthography: fixed, 8-bit – proprietary ASCII (Macintosh OS).

Examples: about 100-1000 complete descriptions of a lexical entry with regard to pragmatics and semantics.

Status:

- demonstration.
- small research.
- stable.
- ongoing development.

Documentation: System documentation: (Grewe et al. 1992): Grewe, K.; Wegmann, F. and Kunze, Cl.: A Tool for Semantics-based Analysis - SEMBLEX. Technical Report. Ruhr-Universität Bochum, Sprachwiss. Inst., 1992.

Upgrades: available.

SourceCode: none.

Consulting: none.

Format: StuffIt Archive.

Price: free.

Restrictions: limited to only personal use for the purpose of research.

Contact: Frank Wegmann.

Address: Ruhr-Universität Bochum / Sprachwiss. Institut / Universitätsstr. 150 / (P.O. Box 10 21 48) / 44801 Bochum / Germany

Telephone: +49-234-700-2461

Email: wegmann@ruba.rz.ruhr-uni-bochum.de (apple-link :WEGMANN.F)

Term Rewrite System for non-confluent TRS's

License: research.

Authors: Wilhelm Weisweber.

Task:

- structure-to-structure transducer (DAG-to-DAG).
- syntactic analysis.
- semantic analysis.
- conceptual analysis.
- transfer.
- generation.

Description: The interpreter for non-confluent term-rewrite systems is the kernel of an experimental machine translation system and performs all structural transductions. The user-defined term-rewrite systems are pre-processed such that the rules can be interpreted efficiently.

Components: term-rewrite rule interpreter.

data: term-rewrite rules.

Modularity: it is one module.

data components independent of program.

Extensibility: omly extensible by the developer.

data can be defined by a computational linguist or linguist who is familiar with GPSG, FAS and BACK system with the help of a structure oriented editor.

Size:

data:

- 10 term-rewrite rules for syntactic analysis (example, German).
- 134 term-rewrite rules for semantic analysis (German).
- 37 term-rewrite rules for conceptual analysis (German).
- 248 term-rewrite rules for transfer (German to English).
- 182 term-rewrite rules for generation (English).

Implementation: Arity Prolog (Editor), Quintus Prolog 3.1 (Interpreter).

Platform: MS-DOS 3.31, AT compatible PC (Editor). UNIX 4.1, Sun workstation .

Languages: German, English.

Retargetability: theoretically every natural language.

Orthography: first order terms (Prolog terms).

Examples: about 100 sentences transduced.

Status:

- small research.
- stable.
- no continuing development.

Documentation: W. Weisweber "Transfer in Machine Translation by Non-Confluent Term-Rewrite Systems" Procs. GWAI-89, Eringerfeld 1989, p. 264-269.

W. Weisweber, Ch. Hauenschild "A Model of Multi-Level Transfer for Machine Translation and Its Partial Realization" KIT-Report 77, Institute for Software and Theoretical CS Technical University of Berlin 1990 and to appear in: Procs. seminar "Computers & Translation '89", Tiflis 1989.

W. Weisweber "Term-Rewriting as a Basis for a Uniform Architecture in Machine Translation" Procs. Coling-92, Nantes 1992, p. 777-783.

System and user documentation in progress.

Upgrades: none.

SourceCode: none.

Consulting: available.

Format:

- ftp.
- 3 1/2 " disks.

Price: free.

Restrictions: none.

Contact: Wilhelm Weisweber.

Address: Technical University of Berlin / Department for Software and Theoretical Computer Sciences / KIT / Sekr. FR 5-12 / Franklinstr. 28/29 / 10587 Berlin / Germany

Telephone: +49-30-314-24928 / -27778

Email: ww@cs.tu-berlin.de

TFS (Typed Feature Structure) system

License: research.
Authors: Martin C. Emele and Rémi Zajac.

Task: The TFS system has been developed to provide a computational environment for the design and implementation of formal models of natural language. The TFS formalism is designed as an executable specification language that can be used for the modelling of a variety of linguistic description paradigms; smaller fragments have been built that were ported from descriptions in DCG, PATR-II, HPSG, LFG or SFG into TFS. The main applications where the system has been used are the development of large HPSG grammars.

Description: The main purpose of TFS is the representation and constraint-based processing of lexical and grammatical knowledge in NLP.

TFS is designed as a specification language for the description of linguistic phenomena. In particular, this representation language is declarative and executable at the same time. Its declarativity supports the representation of linguistic facts independent of a particular mode of execution. Its executability is based on the fact that the objects of the data description language are at the same time the data objects on which the processing machinery operates.

The main features of TFS can be summarized as follows:

- Support for the processing of partial information, as it is being used in unification-based grammar formalisms.
- Support for the construction of specialization hierarchies of types (sets of objects) and the use thereof for the non-redundant representation of linguistic knowledge. This property corresponds to the classifier of knowledge representation formalisms such as the members of the KL-ONE family. It is strongly influenced by the object-oriented programming paradigm used in Artificial Intelligence applications.
- Wellformedness conditions for linguistic objects are expressed via the use of constraints. The use of algorithms from the constraint logic programming paradigm makes these techniques available in a new context for the representation and processing of linguistic knowledge. The utilization of recursive constraints is one of the major distinguishing features of our approach.

Components:

- compiler.
- interpreter/constraint solver.
- graphical and menu-based interface.

data: HPSG-based grammars.

Modularity: none.

data components independent of program.

Extensibility: only extensible by the developer.

data extensible by:

- the linguist.
- the computational linguist.
- the experienced user.
- the new user.

Size:

- 50000 lines of source code.
- 2,2 MB of executable.

Implementation: Common Lisp, TFS language.

Platform: UNIX, MacOS .

Languages: -.

Retargetability: no restrictions imposed by the system.

Orthography: ASCII.

Examples: about 2000 words, phrases and sentences tested.

Status:

- demonstration.
- small research.
- stable.
- ongoing development.

Documentation: Zajac, Remi (1992) 'Inheritance and Constraint-Based Grammar Formalisms.' Computational Linguistics 18, pp. 159 - 180.

User and system documentation: User manual and EBNF syntax description

Upgrades: available.

SourceCode: none.

Consulting: none.

Format: binary distribution as a compressed tar file for the unix version, binhex self-extracting stuffit archive for the Mac version (by anonymous ftp from the address ftp.ims.uni-stuttgart.de [current IP address 141.58.127.8]).

Price: none.

Restrictions: non-commercial use.

Contact: Martin C. Emele.

Address: Universität Stuttgart / Institut für maschinelle Sprachverarbeitung (IMS) / Azenbergstrasse 12 / 70174 Stuttgart / Germany

Telephone: -.

Email: emele@ims.uni-stuttgart.de, tfs@ims.uni-stuttgart.de

12 Data sets

FJGH–grammar

License: research.
Authors: Hiroshi Sano.

Task: linguistic analysis.

Description: 1. Overview

LUG, Localized Unification Grammar, is a phrase-based grammar description formalism that aims at unification grammar formalism.

2. Function

The LUG is a grammar description framework designed to allow users to develop non-trivial grammar rules expresed in the DCG. In the LUG form, categories are represented as feature sets. This allows the users to write a complex constituent structure expressed in the grammar rules.

3. Application

The LUG formalism has been used to build grammar rules for basic coverage of the Japanese language. As of now, a grammar with 800 rules is usable. We call this the FJGH grammar rules.

An important characteristic of the basic grammar is that it is classified into 12 groups by linguistic phenomena. The grammar rules grouped acoording to coverage as follows:

- Elementary Level(1-4):
  - decision(declaratives).
  - supposition.
  - conjectural form(declaratives).
  - command(imperative).
  - aspect operators.
  - negation.
  - polite form.
  - ...
- Intermediate Level(5-8):
  - passives.
  - causatives.
  - modal adverbs.
  - spacio-temporal adverbs.
  - topicalized phrases.
  - relatives.
  - ...

- Advanced level(9-12):
  - conditional phrases.
  - causal phrases.
  - some connectives.
  - conjunctions and disjunctions of nominal phrases.

Components: data:

- grammar rules.
- dictionary.

Modularity: data components independent of program.

Extensibility: data extensible by:

- the developer.
- the computational linguist.
- the experienced user.

Size: data:

- 660 KB data.
- 800-sized grammar rules.
- 30,000 words of dictionary.

Implementation: no special programming language, since these are only grammar rules.

Platform: independent.

Languages: Japanese.

Retargetability: none.

Orthography: 16-bit EUC code (Including KANJI code).

Examples: sentences tested.

Status:

- large research.
- stable.
- ongoing development.

Documentation: User documentation: FJGH / Grammar Rules Reference Manual (In Japanese).

Upgrades: none.

SourceCode: available.

Consulting: available.

Format: -.

Price: free.

Restrictions: none.

Contact: Hiroshi Sano.

Address: Kansai Research Laboratory, Toshiba Corp. / Information Processing Group / Minami-machi / Motoyama / Higaishinada-ku / Kobe city / 658 / Japan

Telephone: +81-078-435-3551

Email: sano@krl.toshiba.co.jp

PC–KIMMO definition files for turkish morphology

License: research.

Authors: Kemal Oflazer.

Task: linguistic analysis.

Description: Turkish is an agglutinative language with complex word formations. We use the system as morphological analysis component in LFG / ATN parsing of Turkish sentences and in text tagging.

Components: morphological analyzer/generator (pc-kimmo).

data: Full scale root lexicon for Turkish root words along with some syntactic feature.

Modularity: see pc-kimmo.

data independent of program.

Extensibility: see pc-kimmo.

data components easily extensible.

Size: see pc-kimmo

data: -.

Implementation: see pc-kimmo.

Platform: see pc-kimmo.

Languages: Turkish.

Retargetability: Turkish, can possibly be adapted to other Turkic languages with minor to major effort.

Orthography: see pc-kimmo.

Examples: about 1000000 examples tested.

Status:

- large research.
- stable.
- ongoing development.

Documentation: User documentation: PC-KIMMO has its own documentation, there is some minimal documentation for the lexicon and rules files for Turkish. There are papers available via anonymous ftp that describe the implementations.

Upgrades: -.

SourceCode: -.

Consulting: -.

Support: -.

Format: UNIX compressed tar file. (can also be used on a Macintosh with 4-6 meg of memory)

Price: -.

Restrictions: non-commercial use.

Contact: Kemal Oflazer.

Address: Bilkent University/ Computer Engineering and Information Science/ Bilkent Ankara, 06533/ Turkey

Telephone: +90-4-266-4133 Fax: +90-4-266-4126

Email: ko@hattusas.cs.bilkent.edu.tr or ko@trbilun.bitnet

13 Applications and text processing

ESTEAM (ESPRIT 316)

License: commercial.
Authors: Thomas Grossi.

Task:

- parsing.
- generation.
- understanding.

Description: The module allows the user to enter pseudo-natural language sentences via menus. Output is a semantic representation of the user's sentence in Functional Descriptions. The same system allows the generation of pseudo-natural language sentences from these Functional Descriptions. The set of input sentences and output sentences can be the same. The system is very easy to extend.

Components: a module for pseudo natural language input and output via menus.

Modularity: independent module.

Extensibility: easily extended.

Size: driver 512 lines; menu defs 1558 lines.

Implementation: Prolog.

Platform: UNIX, Sun.

Languages: English.

Retargetability: yes.

Orthography: -.

Examples: 100-1000.

Status:

- completed.
- large research.

Documentation: Esteam deliverables 16 and 22; a paper currently submitted to various conferences.

Upgrades: not provided.

SourceCode: not provided.

Consulting: not available.

Support: Since Cap Gemini in general sells services rather than software, each case must be considered individually.

Format: Since Cap Gemini in general sells services rather than software, each case must be considered individually.

Price: Since Cap Gemini in general sells services rather than software, each case must be considered individually.

Restrictions: Since Cap Gemini in general sells services rather than software, each case must be considered individually.

Contact: Thomas Grossi.

Address: Cap Gemini Innovation / 7, ch. du Vieux Chene / 38240 / Meylan / France

Telephone: +33-7676-4723

Email: grossi@capsogeti.fr

How to Use IT (Mac)

License: commercial.

Authors: Gary Simons, John Thomson.

Task: glossing (semi-automatically) of analyzed texts.

Description: IT (Interlinear Text) is a specialized editor intended to give linguists, literary scholars, anthropologists & translators a tool for developing a corpus of annotated interlinear text. IT views text as a sequence of text units, each of which contains a text line plus a multidimensional set of annotations provided by the analyst.

Components: -.

Modularity: embedded.

Extensibility: no.

Size: -.

Implementation: C.

Platform: Macintosh.

Languages: any left-to-right script.

Retargetability: see above.

Orthography: -.

Examples: 1000-10000.

Status:

- completed.
- stable.
- production quality system.

Documentation: Gary F. Simons and John V. Thomson. How to use IT: Interlinear Text Processing on the Macintosh. 1988.

Upgrades: not provided.

SourceCode: not provided.

Consulting: available (phone).

Format: Mac disk sold with documentation.

Price: US$199.95.

Restrictions: commercial product- may not be copied or shared.

Contact: Phil Payne.

Address: Linguist's Software / P.O.Box 580 / Edmonds, WA 98020-0580 / U.S.A.

Telephone: +1-206-775-1130

Email: -.

How to Use IT (MS-DOS)

License: research.

Authors: Gary Simons, Larry Versaw.

Task: glossing (semi-automatically) of analyzed texts.

Description: IT (Interlinear Text) is a specialized editor intended to give linguists, literary scholars, anthropologists & translators a tool for developing a corpus of annotated interlinear text. IT views text as a sequence of text units, each of which contains a text line plus a multidimensional set of annotations provided by the analyst.

Components: -.

Modularity: embedded.

Extensibility: no.

Size: -.

Implementation: C.

Platform: MS-DOS.

Languages: any using Roman-based script.

Retargetability: see above.

Orthography: -.

Examples: 100-1000.

Status:

- almost completed.
- large research.

Documentation: Gary F. Simons and Larry Versaw. How to use IT: A Guide to Interlinear Text Processing. Summer Institute of Linguistics, version 1.1. 1988.

Upgrades: provided.

SourceCode: not provided.

Consulting: not available.

Format: 360K or 720K DOS disks included with purchase of documentation.

Price: US$60.00, includes free upgrade to v. 1.2.

Restrictions: none, if used for non-commercial purposes & credit given.

Contact: Academic Bookcenter.

Address: Summer Institute of Linguistics / Academic Computing / 7500 W. Camp Wisdom Rd. / Dallas, TX 75236 / U.S.A.

Telephone: +1-214-709-2404

Email: linda@txsil.lonestar.org

ILA multilingual toolkit

License: commercial.

Authors: Glenn Adams.

Task: multilingual text processing.

Description: General multilingual text processing subsystem independent of language, character set, and platform. Capable of supporting input, processing, and rendering of all the world's languages.

Components:

- text representation.
- text processing.
- text input.
- text rendering.
- tableware.

data: 35 scripts, 20 character sets, 30 keyboards, 10 collation tables, 4 input/output devices (fonts are separate).

Modularity: Toolkit C-library, Tableware Compiler available as independent modules.

Extensibility: can be extended by programmer.

data extensible by the experienced user.

Size: source: 50,000 lines of C code.

Implementation: Ansi-C.

Platform: UNIX, VMS (supports X Windows, Sunview, Character Terminals, Postscript).

Languages: any.

Retargetability: see above.

Orthography: -.

Examples: -.

Status:

- Beta test.
- continuing development.

Documentation: Concepts Guide, API Reference Guide.

Upgrades: provided.

SourceCode: available.

Consulting: available.

Format: QIC tar format.

Price: $10,000.00 Commercial Developer License/ $10,000.00 Academic Research Licence (includes source code and documentation).

Restrictions: none.

Contact: Mark Son-Bell.

Address: International Lisp Associates, Inc. / 114 Mount Auburn St., Cambridge, MA 02138 / U.S.A.

Telephone: +1-617-576-1151

Email: mlt-info@ila.com

KOREKTOR 2.0

License: commercial.

Authors: Jon Jajic.

Task: Spelling checking (coverage: 100,000 headwords) for any textual applications (TSR [resident]-style; suitable for most word processors, databases, etc.).

Description: Part of project - classifier of Czech words into morphological classes (available 10/90) (for dictionary extensions). Dictionary (main; 100,000 headwords) available for research purposes free of charge.

Components: morphological analyzer.

Modularity: single product (firmly embedded).

Extensibility: with difficulty.

Size: working size: 400K, resident memory size 100 K.

Implementation: C, Pascal, ASM.

Platform: MS-DOS, 3.1 and up.

Languages: Czech.

Retargetability: every language = new dictionary.

Orthography: -.

Examples: 1000-10000.

Status:

- ongoing development.
- stable.
- production qualtity.

Documentation:

- user documentation (60 pp.).
- project note, Coling '90 proceedings (J. Hajic & J. Drozd), Helsinki, Finland.

Upgrades: provided.

SourceCode: provided for industry, licensed.

Consulting: not available.

Format: 3 1/2 inch (1 disk) or 5 1/4 inch (2 disks 360K).

Price: US$99.00.

Restrictions: licensed, one user/one copy per price unit.

Contact: Jan Jajic.

Address: Hvozdnicka 1049 / 10000 Praha 10 / Czechoslovakia

Telephone: +42-2-7810623

Email: -.

ORFO

License: commercial.

Authors: O. Grigoryev, A. Steinberg.

Task: The whole system is designed as the first block of NLP, especially for inflective languages. The commercial version of it includes

- partial parsing (for verifying word agreement).
- teaching new words to the system (all word-forms).
- spell-checker for Russian.

Description: The system is designed as the first block for NLP, based on the principles of primary analysis of input sentences by means of subject domain structure. It is supposed to be used for choosing a correct representation of input sentences among several ones.
One of the main advantages of the system is the knowledge acquisition module. It is used for teaching new words with their features to the system. The module can be easily expanded for other inflective languages, any other subject domain. It includes verification of subject domain structure and of acquired knowledge.

Components:

- morphological analyzer/generator.
- partial parser.
- knowledge representation module.
- knowledge acquisition module.
- spell-checker for Russian.

data: vocabulary for 120,000 Russian wordstems and a set of vocabulary processing programs.

Modularity: independent modules.

Extensibility: under certain conditions.

Size: version 1.1 - about 1.6 MB on hard disc and 140 K of RAM.

Implementation: C and Assembler.

Platform: MS-DOS (PC-DOS), version 3.00 or later.

Languages: Russian.

Orthography: -.

Retargetability: Other languages can be substituted easily, the main advantage of the system is an easy way to manage with inflective languages.

Examples: greater than 10,000; commercially circulated.

Status: Version 1.1 of the system completed, version 2.0 is ongoing.

Documentation:

- user's guide.
- linguistic model description.
- algorithms description.
- program documentation.

Upgrades: provided.

SourceCode: provided under certain conditions.

Consulting: available.

Format: three diskettes (copy-protected).

Price: depends upon the actual set of modules desired.

Restrictions: -.

Contact: O. Grigoryev.

Address: Informatic Enterprise / 7 Ostugeva / 103104 Moscow / Russia

Telephone: +7-095-299-99-04

Email: -.

Parser

License: commercial.
Authors: Prospero Software.

Task: linguistic analysis.

Description: PARSER performs grammatical tagging and parsing of English text. It is designed to aid academic and other researchers in the computer analysis of English corpora. Input in the form of a plain (ASCII) text file is analyzed a sentence at a time, and a verticalized output file is generated, in which each word or punctuation symbol is classified with one of about 100 part-of-speech tags, the labels used to represent the tagset being partially reconfigurable by the user. At the core of the program is a chart parser integrated with a very large phrase-structure grammar of English. The grammar references a two-tier lexicon with several hundred thousand entries. In particular, the 70,000 or so most common words in the language are given detailed coverage, including some semantic categorization, thereby minimizing the errors which stem from reliance upon morphological analysis alone. Tagging accuracy typically exceeds 98

Components:

- morphological analyzer / generator.
- parser / generator.

data:

- CFPSG rules.
- English lexicon.

Modularity: There are two versions of the program available, which generate respectively (a) the part-of-speech tag for each word, and (b) the most likely parse tree for each sentence.

data components mainly embedded in the program.

Extensibility: program and data only extensible by the developer.

Size: system requires 500K free RAM and 2Mbytes hard disk space.

data:

- over 1,000 CFPSG rules incorporated.
- over 400,000 entries in English lexicon incorporated.

Implementation: Pascal + Assembler.

Platform: IBM PCs and compatibles, with DOS.

Languages: English.

Retargetability: -.

Examples: 4,000 sentences tested.

Status:

- production quality.
- stable.
- ongoing development.

Documentation: User manual.

Upgrades: available.

SourceCode: -.

Consulting: -.

Support: Written and phoned user queries are answered. Contact developers for source code. Contact developers for consulting.

Format: disk.

Price: A single-user licence for the version generating part-of-speech tags is US595.00. $A single-user licence for the version which can also generate parse trees is US995.00$. Prices for customization: apply to authors. Multiple-user/site licences: apply to authors.

Restrictions: The program and data are copyrighted. The single-user licence is for use on one machine at one time.

Contact: Dr. Mike Oakes.

Address: Prospero Software Ltd / 190 Castelnau / London SW13 9DH / England

Telephone: +44-81-741-8531

Email: prospero@prospero.demon.co.uk

STAMP

License: research.

Authors: David Weber, H. Andrew Black, Stephen R. McConnel., Alan Buseman.

Task: machine translation.

Description: STAMP is a computer tool used in conjunction with AMPLE to adapt one language to a closely related dialect.

Components: -.

Modularity: -.

Extensibility: -.

Size: -.

Implementation: C.

Platform: UNIX, MS-DOS.

Languages: any.

Retargetability: see above.

Orthography: -.

Examples: 10-100.

Status:

- ongoing development.
- small research.

Documentation: STAMP: A Tool for Dialect Adaptation.

Upgrades: not provided.

SourceCode: provided.

Consulting: not available.

Format: DOS diskette.

Price: -.

Restrictions: none, if used for non-commercial purposes & credit given.

Contact: Academic Bookcenter.

Address: Summer Institute of Linguistics / Academic Computing / 7500 W. Camp Wisdom Rd. / Dallas, TX 75236 / U.S.A.

Telephone: +1-21-709-2404

Email: linda@txsil.lonestar.org

STEMMA

License: research.

Authors: Kimmo Kettunen.
Task: generation.

Description: The program is an implementation of Finnish noun stem generation using a very concrete substring based approach. The product as such is only useful for large scale study of different stem alterations and amount of stems a noun may have. It could also work as a part of an information retrieval system (for generation of the search key prefixes). It could also be easily adapted to full wordform generation for language learning and other purposes. Major advances of the program are its robustness and reliability.

Components: morphological analyzer/generator (generator of Finnish noun stems).

Modularity: one integrated program.

Extensibility: -.

Size: 55K (ICX-code, source 32K).

Implementation: ICON (version 7.5, fully compatible with 8.0).

Platform: MS-DOS.

Languages: Finnish.

Retargetability: no.

Orthography: -.

Examples: 1000-10000.

Status:

- ongoing development (finetuning and debugging going on at the end of October 1990).
- stable.
- production quality.

Documentation: At present only in Finnish (a short leaflet). Upon need and request a more comprehensive documentation available at least in Finnish, possibly also in English.

Upgrades: not provided.

SourceCode: provided.

Consulting: not available.

Format: diskette (3.5 or 5.25 inches).

Price: US$20.00 for executable version, source code negotiable.

Restrictions: freely usable for research and teaching; other uses negotiable.

Contact: Kimmo Kettunen.

Address: Kapylankuja 3 B 17 / SF-00610 Helsinki / Finland

Telephone: +358-0-793-293 (home)

Email: kktk_kotus@cc.helsinki.fi

WORDSURV

License: research.

Authors: John Wimbish.

Task: manipulating and analyzing wordlists from language surveys.

Description: WORDSURV provides a tool for field linguists involved in doing language surveys. It is intended for use with portable PC's for actually entering and analyzing data while on site.

Components: phonological analyzer/generator.

Modularity: embedded.

Extensibility: with difficulty.

Size: 100K.

Implementation: C.

Platform: MS-DOS.

Languages: any.

Orthography: -.

Retargetability: see above.

Examples: 10-100.

Status:

- completed.
- small research.

Documentation: WORDSURV: A Program for Analyzing Language Survey Word Lists, 108 pp.

Upgrades: not provided.

SourceCode: not provided.

Consulting: not available.

Support: none.

Format: 720K or 360K MS-DOS diskette.

Price: US$11.00, includes book + diskette.

Restrictions: none, if used for non-commercial purposes & credit given.

Contact: Academic Bookcenter.

Address: Summer Institute of Linguistics / Academic Computing / 7500 W. Camp Wisdom Rd. / Dallas, TX 75236 / U.S.A.

Telephone: +1-214-709-2404

Email: linda@txsil.lonestar.org