

Contents

1. Introduction	5
1.1 The manual	5
1.2 What INTEXT can do	6
1.3 The philosophy of INTEXT	7
1.4 Help for old friends – what’s new in version 4.0	8
1.5 Quick installation for experts	10
2. Installation and configuration	11
2.1 The automatic installation with ITXINST	11
2.2 INTEXT 4.1 under Win9x	12
2.3 Working with projects	12
2.4 INTEXT in networks	13
2.5 Configuration of INTEXT	13
2.6 External programs to use with INTEXT	13
2.7 The files of INTEXT	14
2.7.1 The provided files	14
2.7.2 The system of file names	15
2.8 Hardware configuration	17
2.8.1 VGA-Graphic adaptor	17
2.8.2 Printer	18
2.8.3 The Use of Expanded Memory (EMS or XMS)	20
2.9 Problems with installation and configuration	21
2.10 Journey through the system	22
3. INTEXT – an overview	25
3.1 The input files	25
3.2 The output files	25
3.3 The supervisor IS	25
3.3.1 Edit	27
3.3.2 Exploration	27
3.3.3 Analysis	30
3.3.4 Management	32
3.3.5 Results	33
3.3.6 Project	35
3.3.7 Options	36
3.4 The menus	37
3.5 The help system	38
3.6 The line editor	38
3.7 Interfaces to other programs	39
4. Preparing the text	41
4.1 Rules for the definition of external variables	42
4.2 Examples: text units and external variables	43
4.3 Converting of machine readable text data	44

4.4	Generation of a system file – ISYS	47
4.4.1	Rules for writing	47
4.4.2	Rules using variable format	48
4.4.3	Rules for using fixed format	53
4.4.4	Rules for using line format	53
4.4.5	Rules for using paragraph format	53
4.4.6	Rules for using page format	53
4.4.7	Rules for processing the SOLIS data bank	54
4.4.8	Rules for processing the SRM data bank	54
4.4.9	Parameters of the program	55
4.4.10	Information messages	56
4.4.11	Printed result of a system file with external variables	57
5.	Definition of samples	59
6.	Analyses of texts	63
6.1	Word lists	64
6.1.1	Normal form	64
6.1.2	Information messages	66
6.1.3	Reverse word list	67
6.1.4	Printed results of a word list (normal form)	69
6.1.5	Printed result of a reverse word list	70
6.2	Word sequences	71
6.2.1	Parameters of the program	71
6.2.2	Information messages	73
6.2.3	Printed results of word sequences	74
6.3	Word permutations	75
6.3.1	Parameters of the program	75
6.3.2	Information messages	77
6.3.3	Printed results of word permutations	78
6.4	Cross references	79
6.4.1	Parameters of the program	79
6.4.2	Information messages	81
6.4.3	Printed results of cross references	82
6.5	Comparision of vocabularies	83
6.5.1	Parameters of the program	84
6.5.2	Information messages	85
6.5.3	Printed results of vocabulary comparisions	86
6.6	Searching within a vocabulary	89
6.6.1	Parameters of the program	89
6.6.2	Information messages	90
6.7	TTR dynamics	91
6.7.1	Parameters of the program	91
6.7.2	Information messages	92
6.7.3	Results of TTR dynamics	93
6.8	The use of search patterns	94
6.8.1	Specifications in the parameter field	94
6.8.2	Strings	95
6.8.3	Word root chains	95

6.8.4	Printed output of a category system	97
6.9	Concordances	98
6.9.1	Parameters of the program	98
6.9.2	Information messages	99
6.9.3	Printed output of a concordance (short form)	100
6.10	Search patterns in the text unit	101
6.10.1	Parameters of the program	101
6.10.2	Information messages	102
6.11	Content analysis	103
6.11.1	Category labels	103
6.11.2	Category system	103
6.11.3	Results of the coding	105
6.11.4	Interactive coding	106
6.11.5	Parameters of the program	108
6.11.6	Information messages	110
6.12	List of uncoded words	110
6.12.1	Parameters of the program	111
6.12.2	Information messages	112
6.13	Test on multiple search patterns	113
6.13.1	Parameters of the program	113
6.13.2	Results of the multiple entry test	114
6.14	Readability analysis	115
6.14.1	Parameters of the program	116
6.14.2	Information messages	117
6.15	Personality structure analysis	118
6.15.1	Parameters of the program	118
6.15.2	Information messages	119
6.16	Qualitative analyses of text	119
7.	Data management	121
7.1	Sorting	121
7.1.1	Parameters of the program	122
7.1.2	Information messages	122
7.1.3	Working with sort order tables	122
7.2	Merging of vocabularies	124
7.2.1	Parameters of the program	124
7.2.2	Information messages	125
7.3	Reducing vocabularies	125
7.3.1	Parameters of the program	126
7.3.2	Information messages	127
7.4	Convertings	128
7.4.1	Converting a reverse vocabulary into normal form	128
7.4.2	Converting a word list into a file of search patterns	128
7.4.3	Convert category system	128
8.	Working with code pages	129
8.1	Using code pages	129
8.2	Coding of non displayable characters	134

9. The INTEXT Result Manager	135
10. Error messages	137
10.1 Errors with hardware	137
10.2 Errors with files	138
10.3 Errors occurring with the menus	139
10.4 Errors during the execution of a program	140
11. The INTEXT programs	143
12. The structure of the INTEXT files	145
12.1 ITX-file: system file	145
12.2 DIC file: search patterns	145
12.3 WB? file: word list, word combinations, word permutations	145
12.4 XRF file: cross references	145
12.5 VEC file: sequence of codes	146
12.6 TAB file: code counter	146
12.7 sss/ssl/sis/sil file: concordances	146
12.8 WCP file: short word comparison	146
12.9 WCP file: long word comparison	147
12.10 TTR-file long: TTR-dynamics	147
12.11 TTR file short: TTR-dynamics	147
13. Bibliography	149
14. Glossary	157
15. Index	161

1. Introduction

1.1 The manual

The text analysis program INTEXT is best for all applications in the humanities and social sciences. The first version was developed in 1983 on the IBM-mainframe (IBM 3032 running under MVS) at the computer centre of the university of Münster/Germany and written in PL/1. The versions for PCs are written in C, using the Symantec C/C++ compiler.

This manual was printed using the typesetting program T_EX and its macro collection L^AT_EX. Computer software and their documentation tend to have gaps, incompatibilities, or errors. If you found some, send your suggestions to:

Dr. Harald Klein
 Social Science Consulting
 Brückengasse 12
 07407 Rudolstadt
 Germany

Tel/Fax: +49 3672 488494

Information sources in the internet:

e-mail	webmaster@intext.de
Homepage	http://www.intext.de
Text analysis software overview	http://www.textanalysis.info
Bibliography	http://www.intext.de/PUBLICAE.HTM

The Windows version of INTEXT is called TextQuest, a free trial version is available at <http://www.textquest.de> in English, German, and Spanish.

1.2 What INTEXT can do

This chapter is a short description of all possible forms of analyses and is especially useful for novices.

- word list: a list of all strings that occur in the text together with their frequency. It can be sorted by frequency or by alphabet (with sort order table), also strings can be excluded due to their length, their frequency and their occurrence in an exception list (list of STOP-words).
- word sequence: like a word list, output are parts of the text containing x words, where x is variable. Word sequences can be sorted by the first or by the last word of the word sequence. If x takes the value 1, a normal word list is generated.
- word permutations: like a word list, each string is combined with each following string as a two word sequence.
- reverse vocabulary: like a word list, but the order of the characters has changed from left to right, so the first character in a normal word list is the last character in a reverse word list. "Intext" reversed is "txetnI".
- search patterns in text unit: output is the search pattern and the complete text unit in which the search pattern occurs.
- cross reference: for each string the locations are the output with all external variables. The locations can be formatted in multiple columns.
- vocabulary comparisons: two word lists, word sequences or word permutations can be compared. Also statistics are computed.
- content analysis: with powerful search patterns for single and multiple word coding, controlling of ambiguity and negation with log files or by interactive coding, adaptable negation algorithm.
- readability analysis: there are 6 formulas for the english and 2 for the german language. The syllable count algorithm is language independent and can be adapted for other languages. The formulas are dependent on the genre and/or the language of the text, and the sentence must be the text unit.
- concordance: the context of search patterns is written to a line with variable length that can be formatted in two ways.
- personality structure analysis: counting of repetitions which are the basis of Mit-teneckers personality structure analysis to tell the difference between schizophrenic and mentally healthy persons.
- style analysis: a special form of a content analysis
- list of uncoded words: a list of all strings that are not used for a coding in a content analysis.
- data management: functions to sort word lists, word sequences, word permutations and concordances.

1.3 The philosophy of INTEXT

INTEXT is a kind of toolbox with a lot of analyses provided. Nevertheless the programs can be used also for other purposes than originally intended and described in this manual. The use of INTEXT and its design to use the facilities INTEXT provides are explained in this chapter.

open system: INTEXT is an open system. The texts are stored in files that can be used by INTEXT-programs, but also by other programs. All files are plain ASCII-files, control sequences are only used for emphasis purposes like bold face and are described in the relevant chapters. The format of the files can be found in chapter 11 on page 143.

external programs: Reinventing wheels is not satisfactory. Therefore it is possible to imbed MS-DOS programs into INTEXT like text processors, editors, statistical programs and sort programs.

file names: Due to its design as an open system quite a number of files are generated. Because losing control is bad, a system of file names is a built-in feature in INTEXT and you are advised to stick to its rules. The advantage is that file names need not to be entered but just can be accepted with a single keystroke. They are derived from the project name.

project name: It is used to generate the file names using the built-in system. All texts belonging to one project are stored in a directory. Therefore the project name may contain drive and directory names. Thus it is possible to work with different options (e.g. sort order or negation orders) in different projects.

information messages: are written to the screen and to a log file. It is not necessary to copy results manually from the screen.

changing of INTEXT-tables: Due to its design as an open system INTEXT can be altered to one's own needs, e.g. the sort order table, the lists of indicators for negations, the syllable table and the indicators for foreign words.

samples: The text file can be processed completely, or a sample can be defined on the basis of external variables. Then only the text units are processed which are selected. The sample has to be defined before performing an analysis. In most analyses one can choose between processing the whole text or the defined sample.

language independence: The text of the menus, the help system, and the results are stored in the file `intext.mmt`, a pure ASCII-file. It can be changed, e.g. translated into another language. The `intext.mmt`-file exists in three versions: the current version (`intext.mmt`), the german version (`intextd.mmt`), and the english version (`intexte.mmt`). Also the sort order table can be adapted to the language of the text, also multiple characters are possible.

1.4 Help for old friends – what’s new in version 4.0

In version 4.0 several changes to former versions were made, also new terms are introduced. Old names for external variables are identifiers, word sequences were word combinations. Word list, word sequences, and word permutations are summarized using the term vocabulary.

- The user interface is completely new. All file names and parameters are shown, they can be changed using function keys.
- a two-level help system was implemented. With F1 the files are explained, with F2 the parameters of the selected analysis.
- The system parameters of INTEXT are stored in file INTEXT.INI instead of INTEXT.DEF.
- vocabularies (word lists and derivatives) have been extended and changed:
 - The STOP-word files were renamed. STOPE.WB is now ENGLISH.STP, STOPD.WB is now DEUTSCH.STP. Both files were extended.
 - Selection criteria can now be enabled for all analyses (exception is the TTR-dynamics). Strings can be excluded from processing due to their frequency, their length, or their occurrence in a STOP-word file. Punctuation marks cannot be excluded separately, they are now part of the STOP-word files.
 - All vocabularies can now be reversed, also cross references.
 - The number of digits of the TTR can be specified in the TTR-dynamics.
- Comparison now is possible with all sort of vocabularies (*.WB?-files), not just word lists.
- All menus in the content analysis were completely redesigned and are much clearer now.
- The building of category systems based on vocabularies was reprogrammed and enables you also to stop work at a certain point and restart from there.

In version 3.0 basic features were changed:

- The external variables can now consist of characters instead of digits only. A variable number between 1 and 50 is possible, each external variable can be up to 10 characters long.
- Line format has only one external variable: the line counter.
- Paragraph format has only one external variable: the paragraph counter.
- Page format consists of two external variables now: page counter and line counter.
- Control sequences used in free format cannot be used in line, paragraph, and page format, because the number of external variables is variable now.
- The sequence numbers used in fixed format have been deleted (punch card age).
- Relative control sequences using free format are no longer possible, each value must be specified. Control sequences like \$3 for incrementation do not work any more. Old raw texts must be changed, otherwise the ISYS program does not work properly and can crash.
- Only two external variables are possible with the two data banks formats.
- The file format of the system file has changed, the system files must be generated again.
- Sampling is organised for all forms of analyses. At first the sample has to be defined, then one is asked before performing the analysis whether to process the whole text or the defined sample.
- STOP-words can now be excluded during the generation of word permutations and word sequences.
- In content analysis the unit of analysis is always the text unit, an aggregation of the numeric results is no longer possible, one must use statistical software instead.
- Interactive coding can be stopped and continued later from this position.
- All menu texts are stored in one file called INTEXT.MMT.
- Cross references are performed with WORDBOOK now, the CROSSREF-program has been deleted.
- The file INTEXT.DEF has been renamed to INTEXT.INI.
- The support of the lemmatization program LEMMA2 has been withdrawn.
- The batch mode has been removed.

1.5 Quick installation for experts

The quick installation is for MS-DOS experts who have a good knowledge of the operating system. More detailed information can be found in the next chapter.

- Hard- and software requirements
 - space: 1.6 MB on hard disk for programs, drivers and example files
 - RAM: at least 2 MB for small files
- Installation:
 - Run ITXINST, it checks the configuration of the system, creates a directory for INTEXT, copies INTEXT into this directory, configures INTEXT to the VGA graphics adaptor and the printer. Occasionally the file CONFIG.SYS has to be altered:
 - CONFIG.SYS: set files=32 (or higher values)
 - CONFIG.SYS: include the file ANSI.SYS (or the substitute that came with your VGA-adaptor) as a DEVICE
 - configure EMS (without Windows) resp. XMS (with Windows). Attention: the Windows versions of HIMEM.SYS and EMM386.EXE must **not** be used! 386MAX or QEMM work, but only the newer versions. More information can be found in chapter 2.8.3 on page 20.
 - The environment variable INTEXT will be set in the AUTOEXEC.BAT file; the statement is written to the end of this file.
 - have a boot diskette ready or generate one if you don't have one.
 - perform a warm boot or a reset.
 - Work through chapter 2.8.3 on page 21 "Journey through the system".

2. Installation and configuration

The INTEXT-diskette contains the installation program ITXINST, README with the latest information and an ITX*.EXE file, depending on the version of INTEXT (e.g. ITX-386.EXE for INTEXT/386). Make at least one backup copy of the diskette provided. Installation and configuration can be done with ITXINST without problems, but if problems occur, the next chapter contains the information needed to do that "by hand".

2.1 The automatic installation with ITXINST

ITXINST copies the desired INTEXT-version from the diskette into the desired directory on the desired drive, and checks the configuration in the CONFIG.SYS file. If errors are found, warning will be issued. After that the VGA-adaptor and the printer are installed.

The following configuration features in file CONFIG.SYS are checked:

- the number of the open files at one time must be 20 or more: FILES=32
More than 32 have no effect.
- the driver ANSI.SYS – or an extended driver – must be included as a device
- the driver HIMEM.SYS must be included
- if the driver EMM386 is included, the key word NOEMS may **not** be used, otherwise INTEXT cannot use EMS/XMS-storage and will crash.

→ ITXINST cannot recognize whether the files HIMEM.SYS and EMM386.EXE are DOS or Windows versions.

The INTEXT environment variable is set in the AUTOEXEC.BAT-file at the end of the file.

2.2 INTEXT 4.1 under Win9x

Running INTEXT under Win9x is dependent on the configuration of the system, because INTEXT uses a built-in DOS-extender. The examples in the INTEXT-manual only apply for systems with Win3.1 and 32 installed RAM or less. With Win9x only specify HIMEM.SYS and EMM386.EXE in the CONFIG.SYS file as described underneath, then INTEXT will run without problems on systems with more than 32 MB installed RAM.

```
DEVICE=C:\WINDOWS\HIMEM.SYS
DEVICE=C:\WINDOWS\EMM386.EXE
```

2.3 Working with projects

INTEXT must be stored in a separate directory to be able to work with the program. The texts should be stored in another directory, INTEXT is invoked from this directory. To make this work, the following steps must be performed:

- define the environment variable INTEXT, this is done by ITXINST, the SET-command is appended to the file AUTOEXEC.BAT. The SET-command for drive N: and directory INTEXT is:

```
set INTEXT=N:\INTEXT\
```

Important: the \ at the end of the line must exist if a directory is specified, otherwise the file INTEXT.MMT cannot be found (file intext.mmt is missing), and the programs cannot be executed.

- include INTEXT in the environment variable PATH, this is done with a SET-command in the file AUTOEXEC.BAT.

The following files must be copied to each directory where texts that belong to one project are stored:

- devices for printer or VGA-card *.CFG
- printer driver PRINT.DEF
- word lists, word sequences, word permutations, cross references, and vocabulary comparisons: sort order table ISM.DEF
- stop word files *.STP (DEUTSCH.STP for german, ENGLISH.STP for english, and FRANCAIS.STP for french)
- content analysis: the files of negation indicators NEG*.DEF

- readability analysis: current syllable count table REFO.SYL, REFOD.SYL (german), REFOE.SYL (english) and foreign words table FWORTE.DAT

2.4 INTEXT in networks

Within networks INTEXT is to be installed on the fileserver. INTEXT can only be used from the connected PCs when the environment variable INTEXT is set. This should be done in the AUTOEXEC.BAT-file. The *.DEF, *.CFG, *.DAT, *.WB, *.STP and *.SYL files must be copyable (copy permissions has to be set), or there must be copies of these files on local drives of the PCs. The knowledge of users has to be taken into account.

2.5 Configuration of INTEXT

The submenu in the menu allows the configuration of INTEXT. The following features can be configured:

- beep after the end of every program on or off
After the end of every program a beep sounds that can be switched on or off. It's valid for every program.
- line counter on or off
A line counter runs in every program – except IRM – in the left bottom corner on the screen that can be also switched off. Sometimes – on not 100 % compatibles – that may be necessary.
- monochrome or colour monitor connected
Independent from the installed graphic adaptor you can select monochrome or colour display depending on the existing hardware.

2.6 External programs to use with INTEXT

From the supervisor external programs that run under MS-DOS can be imbedded (no MS-Windows programs). Three statistical packages, a sort program and an editor resp. text processor are possible. The names of the executable files (EXE-files) have to be entered, these files must be callable by MS-DOS (the directories must be included in the PATH environment variable).

- name of the statistical program
The name of the EXE-file has to be entered, e.g. SPSSPC.EXE, SAS.EXE or CONCLUS.EXE.

- name of the sort program
The name of the EXE-file that calls the sort program has to be entered, e.g. SORT.EXE, QSORT.EXE or MSORT.EXE. The sort program of MS-DOS SORT.EXE can only sort files up to a size of 64 KB and uses the sort order of the alphabet that is specified in the COUNTRY-variable (specified in the CONFIG.SYS file). Calling the sort program must be the same as with SORT.EXE, e.g.:
sort unsorted.dat sorted.dat /+49
- name of the text processor or editor
For editing of some files an editor or a text processing program is necessary. You can integrate your favourite one if it can read pure ASCII-files. Just enter the name of the command, e.g. WP, WORD, PE, KEDIT or QEDIT. The program must be found via the environment variable PATH. After this command the file name will be added, after that options for the editor/text processor can be specified.

These options can be changed after each analysis or can be stored with the `store` option permanently. The values of the options are written to the file INTEXT.INI.

The defaults can be changed by editing the file INTEXT.INI using j (for yes) and n (for no) in the first 2 columns of the first line (column 1 for beep, column 2 for line counter).

For training purposes the defaults are useful, because the program is transparent to the user. Processing huge amounts of data time can be saved by switching off the line counter (some programs run faster, especially WORDBOOK).

2.7 The files of INTEXT

2.7.1 The provided files

INTEXT consists of the executable programs (*.EXE-files), the definition files (*.DEF), the initialisation file INTEXT.INI, the program messages in file INTEXT.MMT, the *.CFG-files for printers and VGA-cards, and two *.BAT-files. All these files must be in the current directory, this can be changed with the use of the environment variable INTEXT (see section Working with projects). Otherwise the programs are aborted with the error message: file intext.mmt is missing.

The INTEXT.MMT-file contains the menu texts, the help texts, and information messages for every program in three versions: the current (INTEXT.MMT), the German (INTEXTD.MMT), and the English version (INTEXTE.MMT).

Die *.BAT files ENGLISH.BAT and DEUTSCH.BAT copy the menu texts and information messages into the current file, and so the INTEXT-system can change its language very fast. It is also possible to translate the INTEXT.MMT-file into other languages easily.

The *.SYL files contain the legal combinations of vocals in a language. The REFO program

needs it for counting syllables. This file is a pure ASCII-file.

Die *.DEF files define defaults for one or several programs. Their meaning is as follows:

- INTEXT.INI contains the defaults for the supervisor IS that can be altered in the submenu configuration of the options menu.
- ISM.DEF defines the collating sequences, e.g. Umlauts. The programs ISM, WORD-BOOK, WOBANA and WORDCOMP use this file.
- PRINT.DEF contains the defaults for the page layout if a file is printed.
- NEG-PRE.DEF contains the indicators for negation that are searched before a search pattern. This file is used by SUWACO.
- NEG-POST.DEF contains the indicators for negation that are searched after a search pattern. This file is used by SUWACO.
- VGATMODE.DEF contains the values for the extended text modes for each file that can be output from the supervisor IS in the results menu. This file is used by the IRM program and generated by the ITXINST program.

The PRINT.CFG file contains the control characters for the printer. Default is the NEC Pinwriter P6.

2.7.2 The system of file names

The names of the input and output files are derived from the project name, the following file extensions are used:

- clg: coding control log file
- ctx: coded text units
- def: definition file
- dic: category system (file of search patterns)
- dse: file with multiple search patterns
- fwp: rapport file of foreign words
- ids: definition of the sample for the project
- itx: system file
- lab: label file (category labels)

- log: file with rapport of the results
- ntx: file of negated text units
- ovl: file of overlapping text passages
- prj: file with data of the project (internal use)
- prn: printable/editable file (generated by IRM)
- pro: control of foreign words
- rtx: uncoded text units
- rwb: file of uncoded strings
- sco: control of syllables
- sis: unsorted concordances in short format
- sil: unsorted concordances in long format
- sit: search patterns in text units
- sss: sorted concordances in short format
- ssl: sorted concordances in long format
- sst: sorted search patterns in text units
- tab: file of counters in a content analysis
- ttr: TTR dynamics
- txt: raw data
- vec: file of codes in a content analysis
- wb: word list
- wbc: word sequences
- wbg: vocabulary with GO-words
- wbp: word permutations
- wbr: reverse word list
- wbs: vocabulary without STOP-words
- wcp: comparison of vocaularies
- xrf: sorted cross references

2.8 Hardware configuration

2.8.1 VGA-Graphic adaptor

IRM (Intext **R**esult **M**anager) can display the results on the screen and can use the extended text modes of VGA-adaptors. These extended text modes are **not** standardised, every manufacturer uses different values to switch to different text modes. Some VGA-adaptors do not support text modes that INTEXT requires. So word lists or concordances in long format cannot be displayed correctly.

The installation program ITXINST and the configuration program VGAINST test all available extended text modes, but only work with register compatible VGA-adaptors. If the modes for word lists or long concordances are not available, an error message is issued. The information is written into two files:

- VGAMODES.DOC: contains all different extended text modes of the VGA-adaptor. The three values mean: value of text mode, columns/lines and lines/page.
- VGATMODE.DEF: contains all values of the text modes (decimal) used by IRM.

If both programs do not work, the file VGATMODE.DEF can be generated manually. In that case it's likely that the VGA-adaptor is **not** compatible and IRM will not be able to switch between the text modes. This can be the case with old VGA-adaptors. If ITXINST or VGAINST are used, INTEXT is automatically configured, if not, the configuration can be achieved as described in the following section.

Two things are to do:

- include the extended ANSI.SYS driver into the CONFIG.SYS file. This driver is named e.g. TANSI.SYS for VGA-adaptors with Trident-Chip or EANSI.SYS for VGA-adaptors with ET3000 or ET4000 Chip from Tseng Labs. The driver is on one of the diskettes that came with the VGA-adaptor. The ANSI.SYS file of MS-DOS may not be included, it is substituted by the extended ANSI.SYS driver. After a reboot or a reset the driver will be activated. If you don't have an ANSI.SYS file in your CONFIG.SYS file, the colours will not work properly. Example for the inclusion of a VGA-adaptor with Trident-Chip into the CONFIG.SYS file:
 - until version 4.01 of MS-DOS inclusive: `DEVICE=TANSI.SYS`
 - version 5.0 (or higher) of MS-DOS or with memory manager: `DEVICEHIGH=TANSI.SYS`
- manual installation. You have to create the VGATMODE.DEF file. The structure is the same as the order of the submenus in the results menu of the supervisor IS:
 1. system file
 2. vocabulary

3. category system
4. short concordance
5. long concordance
6. vocabulary comparision
7. cross references
8. uncoded text units
9. coded text units
10. negated text units
11. uncoded words
12. search pattern in text unit
13. reverse word list
14. vocabulary without STOP-words
15. vocabulary only GO-words
16. search patterns in text unit
17. complete coding control
18. multiple search patterns

The number of the equivalent text mode must be written as **decimal** value in every line. The file is a pure ASCII-file. The following text modes are suggested for the different files:

type of file	columns/line	lines/page	example formats
system files (*.?tx)	80	25	80x25, 80x30, 80x43, 80x50
short concordances	80	25	80x25, 80x43, 80x60, 80x65
long concordances	132	25	132x25, 132x43, 132x60
vocabularies	80	55	80x60 (not 80x43), 132x60
other files	80	25	80x25, 80x43, 80x60, 80x65

Any key pressed will show the next page towards the end of the file, q or e leave the result manager IRM and return to the supervisor.

If the VGATMODE.DEF files does not exist, IRM will ask for the text mode to be switched to. The defaults are for VGA-adaptors with Trident-Chip, all other VGA-adaptors may have other values.

→ all values must be entered as decimal values, **not** as hexadecimal ones.

2.8.2 Printer

If you didn't install a printer with ITXINST, you can do it as follows: INTEXT is delivered with several printer drivers, default is the NEC P6 Pinwriter. The other ones are for the Epson LQ, the HP DeskJet and the Star LC-10 and LC24-10 printers. A simple copy command installs the printer.

The EPSON-LQ.CFG file is a driver for nearly all printers of the EPSON LQ-series, HPDESKJ.CFG for the HP-DeskJet series (DeskJet, DeskJet+, DeskJet 500 and DeskJet 550C) and NEC-P6 for nearly all printers of the NEC-Pinwriter series. All others are model specific.

Example for the installation of an EPSON-printer:

```
copy epson-lq.cfg print.cfg
```

All other printers must be adapted. Nearly every printer can be adapted for IRM. The control sequences (ESC-sequences) are stored in the PRINT.CFG file. Every line consists of a sequence of six codes (decimal values (e.g. ESC=27), separated by blanks. If a code is shorter than six values, the rest up to six has to be filled with a -1. If a feature is not available with your printer, you have to write the -1 six times. Every control sequence has a separate line as follows:

1. control sequence for 10 cpi
2. control sequence for 12 cpi
3. control sequence for 15 cpi
4. control sequence for 17 cpi
5. control sequence for 20 cpi
6. control sequence for the smallest vertical movement (e.g. n/216, than you must enter 216 under item 11).
7. printer initialisation sequence (e.g. IBM-character set using NEC Pinwriter)
8. national character set (without country)
9. control sequence for printing attribute on
10. control sequence for printing attribute off
11. value for smallest vertical movement (e.g. 216)
12. page advance (only for using single sheets: distance between the top paper edge and the first printable line. The page advance is part of the top margin.)
13. control sequence for new page using automatic sheet feeders

The printing attribute (e.g. bold face) defines the emphasizing of the search patterns if you print these. The type of emphasizing is dependent on the printer.

The defaults for the page layout can be stored in the PRINT.DEF file. Every specification is written in one line. The order of the lines is as follows:

1. left margin in mm

2. right margin in mm
3. top margin in mm
4. bottom margin in mm
5. paper width in mm
6. paper length in mm
7. line density
8. national character set
9. line spacing (in decimal punctuation)
10. type of paper (single sheet=ja, all other = continuous paper)
11. character spacing (cpi-value)

2.8.3 The Use of Expanded Memory (EMS or XMS)

This chapter applies only to PCs with MS-DOS 5.x and Windows 3.x installed, not for Win9x machines.

EMS-memory is used if you don't use Windows, XMS-memory is used together with Windows 3.x. The file CONFIG.SYS has to be altered if extended memory is to be used. Both kinds of extended memory require a software driver that activates it (e.g. 386MAX, version 5, HIMEM.SYS (provided with MS-DOS since version 4) or QEMM). If EMS-memory is provided with EMM386, you must use the MS-DOS version¹. The available storage can be shown with the program EMS-RAM².

In the first example the available 8 MB RAM are divided into a RAM-disk of 2048 KB and provides EMS-storage of 4784 KB. The extended ANSI.SYS driver TANSI.SYS for a Trident VGA-adaptor is invoked. In line 3 parts of the VGA-storage is defined for use of MS-DOS RAM, the values may be different for other VGA-adaptors.

→ If you test these values, please have a boot diskette at hand or generate one before testing.

¹If the Windows version is used, the program might crash.

²With EMS-storage INTEXT is approx. 8 % faster than with XMS-storage

1. example: 8 MB RAM, 2 MB RAM-disk, rest EMS-storage, no Windows

```
device=HIMEM.SYS /NUMHANDLES=64
dos=high,umb
devicehigh=C:\DOS\EMM386.EXE 4784 FRAME=CC00 I=B000-B7FF I=E000-F7FF RAM
devicehigh=C:\DOS\tANSI.SYS
devicehigh=C:\DOS\RAMDRIVE.SYS 2048 512 128 /E
SHELL=C:COMMAND.COM C:\ /P /E:768
country=49,437,C:\DOS\COUNTRY.SYS
files=32
buffers=30
```

2. example: 4 MB RAM, no RAM-disk, EMS-storage, no Windows

```
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH,UMB
devicehigh=C:\DOS\EMM386.EXE FRAME=D000 2768 RAM
BUFFERS=30
FILES=32
COUNTRY=049,,C:\DOS\COUNTRY.SYS
SHELL=C:\COMMAND.COM C:\ /E:1024/F/P
STACKS=9,256
DEVICEhigh=C:\WINDOWS\MOUSE.SYS /Y
devicehigh=c:\dos\ansi.sys
```

3. example: 4 MB RAM, no RAM-disk, XMS-storage, Windows

```
DEVICE=C:\WINDOWS\HIMEM.SYS
DOS=HIGH,UMB
BUFFERS=30
FILES=32
COUNTRY=049,,C:\DOS\COUNTRY.SYS
SHELL=C:\COMMAND.COM C:\ /E:1024/F/P
STACKS=9,256
DEVICEhigh=C:\WINDOWS\MOUSE.SYS /Y
devicehigh=c:\dos\ansi.sys
```

2.9 Problems with installation and configuration

The automatic installation does not cause any problems in most cases. If this is not the case, read this section.

4 MB RAM are sufficient, configured as EMS or XMS storage. Problems with TSR-programs are not known.

The supervisor and other programs may not work, if the FILES= parameter in the file CONFIG.SYS has a value lower than 20³. If this is the case, existing files are not found or new files cannot be allocated. Error messages like "file doesn't exist" or "new file cannot be allocated" (although there is enough free space of the harddisk/diskette) are hints for this.

The command interpreter (usually the file COMMAND.COM) must be accessible. To achieve that, this file

- must be in the current drive in the current directory or
- in a directory listed in the environment variable PATH (the MS-DOS command SET displays the values of all environment variables) or
- must be specified in the environment variable COMSPEC.

Under no circumstances this file may have the hidden attribute. All programs called from the supervisor can't find the command interpreter then and thus cannot be executed. Details are described in the chapter about error messages.

The following CONFIG.SYS-file contains an example:

```
DEVICE=ANSI.SYS
BUFFERS=20
FILES=32
```

All INTEXT-versions require a storage manager (e.g. HIMEM.SYS). If an EMS-driver (e.g. EMM386.EXE) is present, the key word NOEMS may **not** be used. Additional RAM must be organised as EMS or XMS-storage (see chapter 2.8.3 on page 20). The file ANSI.SYS (rsp. a special version for the graphic card) must be present, otherwise the colours of the menus will not work properly.

2.10 Journey through the system

With the help of the provided example file the most important text analyses can be performed. Follow this guide and you get some experience how INTEXT works. At first you invoke the supervisor with the following command:

is kontakt

³The maximum value is 32, higher values – until 255 – are possible, but have no effect. This applies to all MS-DOS versions including 6.2

At first the INTEXT logo appears, after that the menu bar. Move the cursor to the right, the submenu will show **Generation of a system file**. Now press RETURN, the supervisor now calls the program ISYS. File names need not to be keyed in because the supervisor IS uses a system of file names, at this point the input file is KONTAKT.TXT and the output file is KONTAKT.ITX. The system of the file names is described in chapter 2.7.2 on page 15). The screen shows file names and parameters than can be modified. The function keys are used for further processing. **F 1** shows you the help texts for files, **F 2** explains the parameters, **F 3** is used for altering the parameters, pressing **F 4** starts the analysis, and **F 10** brings you back to the supervisor. Longer input data (e.g. file names) are displayed within a line editor and can be edited. Try it first with the characters that are to be truncated from a word: characters can be inserted or deleted, with POS1 you can jump to the beginning of a line, with END to the end of a line. When the input is finished, just hit the RETURN key, and ISYS starts. In the left corner of the screen you see the line counter. A beep tells you that the program has finished, and you can see the results on the screen. To continue, just hit the RETURN key. There is no need to write down the results, that's already done in a file named log file. For every project the results of the programs are written to this file, here its name is KONTAKT.LOG.

Now you have generated a system file that all other programs need. The next step is to generate a word list.

You returned to the supervisor where you started: in the menu **exploration** and its submenu **system file**. Now move the cursor down until the submenu **wordlist** occurs. Press the RETURN-key, and use the functions key like in the previous step when you generated the system file with the ISYS-program.

The WORDBOOK-program will now split the text into single words and will provide a counter for every different one. The sort order table ISM.DEF is used, so that the correct German sort order of the umlauts is achieved. Also differences in case folding – if two words only differ in upper-/lowercase letters – are ignored. When the program stops, you can see the results. These are also written to the file KONTAKT.LOG. Look at the file names – you will find them at the top of the screen – and you will notice that the file types changed (that are the three letters after the full stop). INTEXT uses a system to generate the file names, so that it is seldom necessary to type in these yourself. The wordlist is stored in the file KONTAKT.WB, the results are written to the file KONTAKT.LOG. After that you find yourself back in the supervisor where you started.

Now you want to have a look at the wordlist. The INTEXT-result manager IRM can show the wordlist on the screen, print it or write it to an output file. Press the R, and the cursor will move to the menu **results**. Move the cursor down to the submenu **wordlist** and press the RETURN-key. At first you are asked how many columns you want to have. 4 is a good choice, but you can enter values between 2 and 6 inclusive. You can now decide where the you want the results. Chose the screen. If you have a VGA graphic adaptor installed you will be asked which mode it should use. Because these numbers are depending on hardware, look in the manual of your VGA-card or read the installation part of this manual. Chose a mode that has at least 55 lines per page, most VGA-cards have a 80x60 (characters per line x lines per page) mode, others have a 132x60 mode. Any key will show the next page, entering Q quits.

The speciality of INTEXT is the content analysis. Just hit the **A** to reach the menu **analysis**, the first submenu is **content analysis**. The SUWACO-program performs the content analysis if you hit the **RETURN**-key. This program differs from all others because it has many parameters. If you press **F 4**, the content analysis starts. If you changed the parameters with **F 3** so that all search patterns are marked as potentially ambiguous, an interactive coding is performed. You can chose for each search pattern, whether the search pattern fits in this category or if it does not. With the **F1**-key you can display the labels of the categories onto the screen. With picture ↓ and picture ↑ you can browse through this file. The statistics of the results are comprehensive.

If you want – and have SPSS/PC+ installed⁴ – you can perform the statistical analysis in form of a frequency tabulation. Move the cursor down to the submenu **statistical analysis**, and it will be done. If you finished SPSS/PC+, you will find yourself back in the supervisor.

That was the journey through the system, and you now have an impression how INTEXT works. More details are in the following chapters.

⁴Only the DOS-version is supported, not the MS-Windows one. You have to configure SPSS/PC+ using the submenu **external programs**.

3. INTEXT – an overview

3.1 The input files

INTEXT is an open system. That means that the single programs write files that other programs read. The programs depend on each other, you find more information in the chapter structure of INTEXT. Only three files must be generated with an editor or a text processor:

- a file with the raw texts
- a file with the search patterns (the category system)
- a file with the category labels

If your raw texts is organised in more than one file, these have to be merged into one single file. This can be achieved with a file copy program (e.g. copy of MS-DOS) or with an editor. The file of the search patterns and the file of the category labels can be generated interactively (menu management, submenu generate category system).

The supervisor makes the knowledge of the internal structure of INTEXT nearly unnecessary, because it calls the necessary programs in the required order and generates the file names automatically.

3.2 The output files

Nearly all programs write their results to one or more files that can be processed with other programs. The file formats can be found in chapter 11 on page 143.

3.3 The supervisor IS

The supervisor IS is the "main menu" of the INTEXT system. Before the applications are discussed in detail, an overview follows.

If you follow the conventions of file names, the supervisor generates all file names automatically. The LOG-file contains the names of all files, so that is always possible to check them.

The project name is the basis for the automatic generation of file names and may contain drive and directories. It is recommended that you specify the project name if you invoke the supervisor:

is kontakt or also

is e:\proj17\kontakt

Since version 5.0 of MS-DOS 5.0 the supervisor IS can also be loaded into high memory if enough space (ca. 85 KB) is available with the following command:

lh is kontakt

If there is not enough memory available, error messages can occur then some lines of the file INTEXT.MMT could not be read. Just leave the supervisor IS and restart:

is kontakt

If the project name is not specified, you are asked for it. The project name is changed any time when the input file name is presented and you overwrite it.

The menu bar of the supervisor is ordered from the left to the right like you will work with the system. If menus are enclosed in < and >, then these programs are planned or under development.

With the cursor keys you can move within the menu bar to the right and to the left, so that you can see the menu items. You can choose them by moving the cursor up or down and highlight the desired item. If you press the first letter of an item of the menu bar, the cursor moves directly to that menu.

3.3.1 Edit

edit	explore	analysis	management	results	project	options	end
<div style="border: 1px solid black; padding: 5px;"> raw text category system category label sort order table negation table syllable table foreign word table page layout printer codes </div>							

All files of the INTEXT system can be edited with an editor or a text processor. To use this feature you have to install the one you use by choosing the menu `options`, its submenu `configuration` and enter the name of this program (e.g. WORD or QEDIT⁵). The editable files are shown in the `edit` menu.

3.3.2 Exploration

edit	explore	analysis	management	results	project	options	end
<div style="border: 1px solid black; padding: 5px;"> generate system file generate word list compare vocabularies short concordances long concordances list of uncoded words TTR-dynamics cross reference list reverse vocabulary category system multiple search entries search entry in text unit word sequences word permutations </div>							

The single programs allow different analyses of the text. The following sections describe them in detail.

3.3.2.1 System file

At first a system file must be generated from the raw text. This system file is absolutely necessary to perform any further analysis.

⁵No MS-Windows program

3.3.2.2 Word list

A word list is a table of all strings within a file and its frequency. It is used both to spot input errors and as a working help for the building of categories in a content analysis. Working with the sort order table ISM.DEF and ignoring difference in upper-/lowercase (case folding) are possible. Also strings can be excluded due to their length, their frequency and/or their occurrence in a STOP-word file.

3.3.2.3 Compare vocabularies

Two vocabularies can be compared. All strings that occur in the second file but not in the first file can be written to an output file. The statistical information messages include inclusive and exclusive strings of both files. Umlauts are processed correct, because the sort order table ISM.DEF is used. The complete comparison can be written in three formats:

- short format: the frequencies of the strings in the first file, the second file, the difference and the string are written. The counters have 9 digits.
- long format: for each file the frequencies and the strings are written, both parts are separated by the difference. The counters have 7 digits, the strings are truncated after 39 characters.
- difference format: for all strings that occur in both files their frequencies, their difference (frequency in 1. file - frequency in 2. file) and the string are written to the output file; frequencies and differences have 9 digits.

3.3.2.4 Concordance – search patterns in context (SIC)

Especially potential ambiguous search patterns require more information about the context in which they occur. One form to achieve this is to allocate a line for each search pattern, where the search pattern is in the middle of the line. With IRM the concordance can be emphasized (e.g. underlined, **bold face** or in *italics*). Details how to achieve this can be found in chapter 2.8.1 on page 18. The length of the context can be varied by retaining or skipping (the context is bigger) the external variables and by the variable line length. The output with IRM can be done in two formats: the short format, where each line is written one after the other, and the long format, where each new search pattern causes a centered headline preceded and followed by a blank line.

3.3.2.5 List of uncoded words

If you perform a content or style analysis with search patterns in infix position (search pattern at any place within a string), it makes sense to have a word list that only contains those strings that were not found by coding. This word list of uncoded strings can be used

to find strings that can be search patterns for an already existing category system. This is done by comparing the category system stored in the file of search patterns and the word list. All strings that are not found with the search patterns are written to the file of uncoded strings, case folding (option U) is active. The word list of uncoded strings can be reduced with the stop word file ENGLISH.STP which is part of INTEXT.

3.3.2.6 TTR dynamics

After each word the current value for the TTR is calculated and written to an output file. These data can be processed with other programs, e.g. those written by Gabriel Altmann. This linguistic application demonstrates the dynamics of the vocabulary of a text. The types can be suppressed in the output file.

3.3.2.7 Cross references

For every string cross references are generated. The WORDBOOK-program writes cross references sorted by alphabet to an output file. Strings can be excluded if they occur in a STOP-word list or by their length, but not by their frequency. You can set dashes between the references optionally and you can specify the number of references per line. In interactive mode you can decide whether you want to have the string cross referenced or not. The strings can be underlined, **bold face** or in *italics* with IRM.

3.3.2.8 Reverse vocabulary

A reverse vocabulary is a vocabulary where all strings are reversed, so that the first letter of a string becomes the last, and the last letter becomes the first letter. "Klein" will be reversed to "nielK". A reversed vocabulary can be converted back to its normal form, useful e.g. to examine endings of words. Also word sequences, word permutations, and cross references can be reversed.

3.3.2.9 Generate category system

A category system for a content or style analysis can be generated in two ways:

- a-priori with an editor or a text processor. The program specified in the Options menu is loaded and the file name is passed.
- a-posteriori on the basis of a word list. The file for the search patterns (DIC-file) and the file for labels (LAB-file) are allocated. For every string of the word list one is asked whether it should be part of the category system. If it is accepted, questions for its code and its label – if the code has not used before – follow. The case folding and ambiguity options can be defined for every search pattern separately.

3.3.2.10 Multiple search patterns

The test on multiple search patterns is necessary to find search patterns in a category for a content or style analysis which occur more than once or a part of another search pattern. These search patterns result in multiple coding and can bias the results of it. The output can also be stored in a file.

3.3.2.11 Search patterns in text unit

Search patterns in text unit are similiar to a concordance. At first the search pattern is written, the whole text unit follows. It is possible to analyse the whole context independent from the line length. Because the search patterns are at the beginning of each line, sorting by search patterns is easy if the sort program can handle lines with a variable length.

3.3.2.12 Word sequences

Word sequences are parts of texts that consist of several words, this number is variable. One can exclude those word sequences that contain a word that also occurs in a STOP word list. Word sequences can find the number of phrases, and it also can be used to define search patterns for a content analysis, or for disambiguation.

3.3.2.13 Word permutations

Word permutations are lists of two word sequences where each string is combined with each following string. One can exclude those permutations that contain a word that also occurs in a STOP word list. Word permutations can be the basis for word root chains in a category system.

3.3.3 Analysis

edit	explore	analysis	management	results	project	options	end
		content analysis readability analysis statistics with SPSS statistics with SAS cluster analysis with ConClus personality structure analysis					

3.3.3.1 Content analysis

The SUWACO-program performs a content analysis. Every search pattern from a category system stored in a DIC-file is searched in the system file. Case folding can be ignored (umlauts and characters with diacritics are treated correct) or not. Single and multiple negations in front of the search pattern are recognised. Also an interactive coding of potential ambiguous search patterns including several rapport files are possible.

3.3.3.2 Readability analysis

The REFO-program works with 8 different formulas, please note that the formulas are language dependent (6 for English, 2 for German) and dependent on the genre of the text. The values are between 0 and 100. The higher the value, the easier it is to understand the text.

3.3.3.3 Statistical analysis

The results of the content analysis can be analysed with ConClus (*.STK), SAS (*.SAS), or SPSS (*.SPS). The SUWACO-program writes the necessary setup to a file, together with a description of the data and a frequency tabulation (frequencies, proc freq).

3.3.3.4 Personality structure analysis

A personality structure analysis by Mittenecker is based on the fact that one can tell the difference between schizophrenic and mentally healthy people by their usage of repetitions. PERSANA counts these repetitions within a text unit. The output file contains all repeats, the repeated strings and their distance in words for further analyses.

3.3.4 Management

edit	explore	analysis	management	results	project	options	end
			sort vocabulary merge vocabulry sort concordances sort/merge reverse vocabulary into normal form vocabulary - > DIC-file vocabulary without STOP-words GO-words in vocabulary convert category system				

The INTEXT system provides the program WOBANA for the manipulation of vocabularies. You can include and exclude strings, merge vocabularies and convert them to other formats.

3.3.4.1 Sort vocabulary

Vocabularies can be sorted ascending and descending by alphabet or frequency. Multiple sort keys are not possible.

3.3.4.2 Merging vocabularies

Huge amounts of texts will require to split the text into samples and to generate a vocabulary for each sample, especially with word permutations. These parts of vocabularies can be merged into one vocabulary of the whole text.

3.3.4.3 Sort concordances

Concordances in both formats can be sorted by alphabet, sort key is the search pattern. Remember to use an external sort program if your file size exceeds 64 KB.

3.3.4.4 Sort/merge

The ISM program is invoked without any assumptions to sort INTEXT-files: word lists, word sequences, word permutations, and cross references.

3.3.4.5 Convert reverse vocabularies into their normal form

Reverse vocabularies are converted into their normal form, thus that the first letter will become the last, the second the one before the last and so on. ("nielK" will be converted to "Klein").

3.3.4.6 Convert vocabulary into search patterns

A vocabulary will be converted into a file of search patterns (DIC-file). The code for every search pattern is set to 1, or it can be numbered consecutively.

3.3.4.7 Vocabulary without STOP-words

Unnecessary strings can be deleted from a vocabulary. The strings to be deleted must be in a file, e.g. ENGLISH.STP.

3.3.4.8 Vocabulary GO-words only

Within a vocabulary one can search for strings or parts of them. The strings can be entered interactively or can be read from a file. The output file contains the strings of the vocabulary that were found by the search patterns.

3.3.4.9 Convert category system

The utility program T2I-DIC.EXE is invoked that converts a category system suitable for TEXTPACK into a file of search patterns for INTEXT.

3.3.5 Results

The IRM (Intext Result Manager) is invoked; it allows to write all files generated by INTEXT to the screen, to a printer or to a file. The following menu shows the files that can be handled:

edit	explore	analysis	management	results	project	options	end
				system file			
				word list (WL)			
				category system			
				short concordances			
				long concordances			
				comparision of vocabularies			
				cross reference list			
				uncoded text units			
				coded text units			
				negatex text units			
				list of uncoded words			
				search patterns in text units			
				reverse vocabulary			
				vocabulary without STOP-words			
				vocabulary only GO-words			
				REFO-syllable control			
				REFO-foreign words control			
				multiple search patterns			
				word sequences			
				word permutations			

3.3.6 Project

edit	explore	analysis	management	results	project	options	end
					project files view log file define sample show sample definition MS-DOS (back with EXIT)		

3.3.6.1 Project files

All files of a project are shown on the screen, together with their size and a description of their contents (e.g. system file, word list).

3.3.6.2 View log file

Each INTEXT-program writes its results to the log file. Here you can browse through this file using the keys of the cursor block.

3.3.6.3 Define sample

All INTEXT-programs that process the system file are able to process the whole file or a sample of it based on the selection of the external variables. Here you can define these selection criteria, they are written to the file SELECT.IDS and used by the equivalent programs. Samples can be drawn on the basis of external variables.

3.3.6.4 Show sample definition

Here you can look at the sample definition currently active.

3.3.6.5 MS-DOS (back with EXIT)

With this options you can leave the supervisor and return to the operating system. You can use every program you like as long as there is enough main storage available. With EXIT you return to the supervisor.

3.3.7 Options

edit	explore	analysis	management	results	project	options	end
						configuration	
						external programs	
						project name	
						language of menu	
						language of text	

3.3.7.1 Configuration

Here you can decide whether a program beeps after it did its job, if the line counter (resp. text unit counter) is running, and if the display should be colour or monochrome (e.g. for notebooks). The configuration data are stored in the INTEXT.INI file. Disabling the line counter makes especially the WB-program faster.

3.3.7.2 External programs

External programs can be called from the IS (INTEXT-supervisor): up to three statistic programs (defined are SPSS/PC, SAS/PC, and ConClus), a sort program (defined is SORT.EXE of MS-DOS), and an editor or text processor (defined is EDIT.EXE of MS-DOS). All programs must be accessible via the PATH-environment variable.

3.3.7.3 Project name

The project name is used to generate the names of the files used by INTEXT. Here you can change the project name which may contain drive or directory specifications.

3.3.7.4 Language of menu

The language of the menu, the help system, and the information messages can be changed, currently only English and German are available.

3.3.7.5 Language of text

The language of text to be analysed can be defined so that the the tables for sort order, negation indicators, and syllable counts are the correct ones. Currently only English and German are available.

3.4 The menus

Each application is selected in the supervisor that calls the appropriate program. Its screen looks like this:

INTEXT/586 4.0 - 2/1997 - 06.02.1997 14:14		
routine: ISYS		
application: generate system file		
	input/output files	
file of raw text:		kontakt.txt
name of system file:		kontakt.itx
	parameters	
treated as single words:		.,;:!?()”’
format of raw text:		free format
F1 help files F2 help parameters F3 change F4 start F10 end		

On top of the screen (grey back ground) you find details of the application and date and time. The rest of the screen is divided into three parts:

- input/output files: here all necessary files for the application are specified, absolutely necessary files are marked yellow. If you do not want a file, the file name has to be erased, after hitting the RETURN key a "no" is displayed. If an error occurs, you must enter a valid file name again.
- parameter: the available parameters for this application are specified. The defaults values are accepted with RETURN, change can be done using the ↓ and ↑ keys. Help for the parameter can be invoked with the F 1 key, after the help text is displayed the cursor moves to the beginning.
- The function keys are defined as follows:
 - F 1: help for the files
 - F 2: help for the parameters, also several pages may occur
 - F 3: change of the paramtters, with the RETURN key you jump to hte next until the last parameter
 - F 4: start analysis, the application is activated with the displayed file names and parameters.
 - F 10: end, the application is aborted, the program returns back to the supervisor IS.

3.5 The help system

The help system is organised in two parts and can be activated with function keys. The first step helps with file, the second with parameters of the selected application. Using files errors can occur because input files could not be found. This may be caused by invalid names, drives, or directories or any combination of it. Output files cannot be allocated, if drives and/or directories do not exist or (with MS-DOS) file names do not obey the specifications of the operating system.

All parameters that do not use the line editor help can be invoked with the F 1 key also. The selected parameter is explained. Hitting any key you return to the original string to its beginning, not to the position where you left it.

3.6 The line editor

To enter data, these can be manipulated with a line editor in most of the cases. The line editor is present, when the requested data are strings or longer numbers.

Ins The insert mode can be toggled, default is **insert on** unless the allowed value does not exceed 1 digit. With **Ins** the insert mode can be toggled.

Del the character under the cursor will be deleted and can not be restored.

Pos 1 The cursor jumps to the beginning of the line.

End The cursor jumps to the end of the line.

cursor left the cursor is moved one character to the left. If the cursor is already at the beginning of the line, it will not be moved.

cursor right the cursor is moved one character to the right. If the cursor is already at the end of the line, it will not be moved. The end of the line is limited by the maximum length of the value, e.g. 63 characters if you enter file names.

All other keys of the cursor block have no function within the line editor.

One can issue commands within the entering of files, if files are missing or the file name is wrong:

- **!** – means that a MS-DOS command is to be executed. This command is executed, then you return to the menu where you started from.
Example: `!dir`

- DOS – using this command (it must be in capital letters) you leave the supervisor IS and return to the operating system. All commands can be executed, but one has to have in mind that the available storage (RAM) is smaller, because the supervisor IS occupies space there. With EXIT one can return to the supervisor IS.
- quit – you leave the submenu and return to the main menu of the supervisor IS.

3.7 Interfaces to other programs

INTEXT can generate setups for the following software packages: SAS/PC, SPSS/PC+, and ConClus (Constrained Cluster Analysis).

4. Preparing the text

INTEXT expects a file that consist of external variables of the text and the text itself. This is the reason that the text – called raw text – is separated into text units that are separated with control sequences that specify the values of the external variables. The meaning of the external variables depend on the goal of the analysis. The units of text and analysis must be identical. Within a text unit no value of any external variable can change.

The following rules must be followed:

- The maximum length of a text unit is 100,000 characters, the limit for the PC-version (since version 2.4) ist 32500 characters. Former versions have a maximum length of 9999 characters.
- The maximum size of a text file is only dependent from the mass storage device available (free space on the hard disk).
- The more external variables are used and the longer they are, the bigger the system file is.
- The system file is the basis for all further text analyses.

If the text consists of several files, these must be copied into one single file. And this is the organisation of a system file:

Figure 1: Organisation of an INTEXTsystem file

variables	1. external var	last external var	text (variable long)
1. text unit			The text starts here.
2. text unit			This sentence may be very long.
3. text unit			Or short.
4. text unit			But not more than 50,000 strings in each text unit.
5. text unit			Otherwise there will be no word list possible.
n. text unit			That's all, folks.

The following decisions must be taken:

- What is the definition of a text unit?
- How many and which external variables are necessary?

The definition of a text unit and its external variables are closely related.

4.1 Rules for the definition of external variables

External variables represent variables of the text, examples follow. One external variable is at least necessary, up to 50 are possible. Each external variable may consist of 10 characters, letters and digits may be mixed. Numeric external variables ease statistical analyses, whereas non-numeric external variables (e.g. words, abbreviations) ease the readability of cross references and concordances. Each external variable must consist of at least one character. The values of each external variable can **not** change within one text unit. The values of the external variables are controlled by control sequences using free or fixed format, all other formats work with predefined external variables.

→ The following rules working with external variables have to be followed:

- up to 50 external variables are possible.
- The values of the external variables are separated by dashes within the control sequence.
- The maximum length of each external variable is 10 characters.
- Each external variable may consist of characters available on the machine (MS-DOS PCs: ASCII, Windows: ANSI), but not allowed are tilde (~), number sign (#), dash (-) and the vertical bar (|, ASCII-value: 124).
- The first control sequence using free or fixed format must consist of initial values for all external variables. Working with the other input formats, you work with predefined external variables that do not allow initialisation and cannot be changed.

4.2 Examples: text units and external variables

Content analysis is an empirical hypothesis testing research method. Therefore the definition of a text unit must follow the hypotheses. The following examples show different applications.

1. example: coding of open ended questions

If more than one open ended question in opinion polls is to be analysed, numbers for the interviewed persons and the questions are necessary, because after the coding the coding results have to be merged to all the other variables. The text unit is the answer to one open ended question. If the questionnaire consists of five questions, five text units for each interviewed person exist. If other variables (e.g. gender, age, place) are taken into account, they also must be used as external variables.

2. example: analysis of newspapers/magazines

The most used text unit analysing printed media is the article. Necessary external variables are the name of the medium, the day of print, and a running number of the article within the issue. Also variables like place or size of the article may be useful.

In an analysis about the coverage of environment issues the following external variables were used: the name of the paper, the date, the column, the page within the column, and typographical specialities like photos, comment etc. (Kramer-Santel 1994).

3. example: readability analysis

Readability analyses can only performed when the sentence is defined as a text unit. Also the implications of the used formulas, e.g. language and text genre, must be taken into account. Only one external variable is absolutely necessary: the sentence counter. If several text sources are to be compared, more external variables must be defined.

4. example: literary science, e.g. style analysis

Literature researchers are often interested in the vocabulary of texts and to which period or genre it belongs to. Text units may be chapters, paragraphs or sentences. A chapter as a text unit may cause problems because the maximum length of a text unit is 100,000 characters (approx. 45 pages). More practical are paragraphs as a text unit, and author, book, chapter, and paragraph are useful external variables.

If a comparison of several books of one author is the objective, the sentence should be the text unit, useful external variables are book, chapter, and sentence. Also the page number can be an external variable, but it might change its value within one text unit, so a page number should indicate where the text unit started.

5. example: news in television

A news item is the suitable text unit for the analysis of television news. External variables are the TV station, the date and the current number of the news item. Also technical variables like length in seconds, photos, and type of presentation (e.g. interview, film) can be external variables. This study was done with INTEXT as a Ph.D. thesis (Klein 1996).

6. example: personal advertisements

If the objective is to find out whether there are differences in gender using personal advertisements and amongst different papers, necessary external variables are the name of the papers, the date of issue, and a running number of the advertisement, also external variables are necessary for the gender of the person who advertised and who is looked for. The last external variable describes whether the person is talking of herself, the person that he/she is looking for, and how the relationship shall look like. The advertisements must be separated into several text units depending on what is described. This study was done with INTEXT in 1988 (Giegler and Klein 1994).

4.3 Converting of machine readable text data

The conversion of data into a machine readable format can be achieved in two ways:

- typing of the text (keyboard)
- converting a text into a format that can be processed by INTEXT

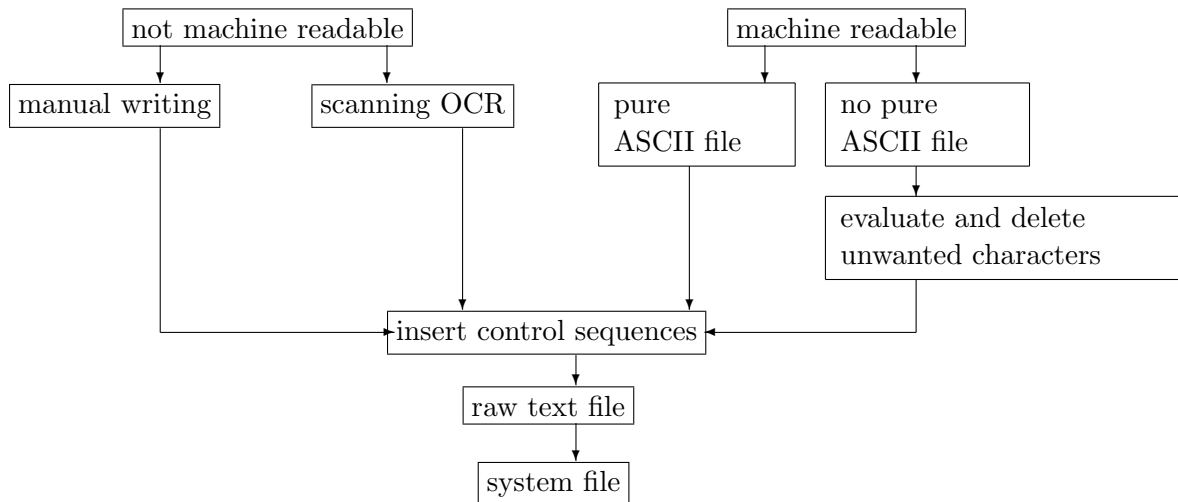
The texts are normally entered via keyboard into the computer or read by a scanner. Manual typing takes a lot of time, and it also requires rules for typing.

The texts can be in several forms:

1. not machine readable (e.g. printed)
2. machine readable as a pure text file (ASCII file)
3. machine readable as a file with unwanted characters (e.g. printing commands)

The following figure shows the working steps between data acquisition and the exploring of the text:

Figure 2: Generation of machine readable texts



If the texts are printed, you can use a scanner and OCR (optical character recognition) software. A scanner works like a photo copying machine, just the image of the page is stored in a graphic format. This file is processed by OCR-software and converted into text. Depending on the quality of the text the characters can be recognised more or less reliable. Good OCR programs have a recognition rate better than 99,9 %, that means each page contains one or two errors. Editing is required, and that has the advantage that one can get familiar with the text.

If the text is stored in a file without printing command, control sequences for the separation of the text into text units have to be inserted.

During the preparation of the text rules for the treatment of characters that do not belong to the English alphabet have to be taken into account. That is a piece of cake with languages that use latin characters. One problem although is the representation of other characters, for example é, É, æ, ô, ò, ì, ÿ or ñ. These problems together with how to use different code pages are discussed in chapter 7.4.3 on page 128.

Languages with a non-latin alphabet, e.g. greek, russian, arabic, or chinese are much more difficult. Other software or working techniques have to be applied. Languages based on syllables (e.g. chinese or japanese) can be coded with multiple characters sets. Other problems are characters with accents or diacritics, especially in french – or language specific characters like ch and l in the spanish language – that are letters. Another point one has to think of is whether typographic variations are important and therefore kept in external variables (e.g. boldface, font size) or in the text.

Another point of interest is pre-editing. That is the marking of phenomena of and in the text with defined character (combinations). An application is e.g. that special categories are to be analysed and these are marked during text preparation. Pre-editing is the most

used working technique for qualitative computer aided content analysis. One reason is that coding with search patterns is based on strings, and not on words in a grammatical sense.

If the text contains phenomena that are important for defining search patterns, one has to make up one's mind how to mark them, e.g. roman numbers which can be words. Also strings that start with numbers but are words might be important. Look at the following examples:

How are football results to be written? 5:2 or 5 : 2 or five two? 5:2 is one word, the second solution contains three words. These has effects on the caluclation of text homogeneity measures like the TTR. Or what about compound words like client/server-technology. Or should it be written client-server-technology? Or client/server technology?

Please have in mind that punctation marks (full stops, exclamation or question marks, commata) follow the words immediately without a blank between them. After punctuation marks a blank must follow because otherwise long words can be the result, and that might cause problems for the software. Hyphenation probably causes problems, please avoid it. Programs like INTEXT cannot distinguish between the hyphenation symbols and dashes.

Printed text today are very seldom typed with a type writer and typeset at the printers, but computers are used for this task. And that means that the texts are already stored in files and could be used. The sources are different, one has to differentiate between text processing and type setting software.

Text processors are mainly PC based (e.g. Word, WordPerfect, Starwriter), type setting systems (e.g. T_EX, DCF, SGML, and HTML) are mostly based on mainframes and UNIX-machines, but that rapidly can change because of the fast innovation in the PC market. Common for both is that there is not only text but also other information for printing stored in the file. These information can be used for the external variables, and after using this information it has to be deleted. Therefore it is necessary that software can convert machine readable texts into a raw text or into a system file format using the information provided to generate text units and external variables.

The text step is the transformation of a machine readable into a raw text format. This is a format that can be converted directly into a system file. INTEXT offers seven raw text formats. Another possibility is the transformation into a system file.

A machine readable text does not mean that it can be converted into a raw text or system file format without some editing work. Control sequences to separate the text into text units and to set the values of the external variables have to be inserted. The next chapter shows how that is done.

4.4 Generation of a system file – ISYS

The ISYS program converts text that is stored in a raw text format into a system file. The raw text formats and their usage are described in this section.

Words in the sense of the program are all characters surrounded by two blanks. Multiple blanks are compressed to one blank while generating the system file.

Some punctuation marks and special characters (e.g. ., ; : () etc.) should be defined as own words. This has the advantage that some strings don't occur more than once in a word list because before or after them there are these characters, and also in a content analysis the search patterns and their coding is not biased, especially if the search patterns are in infix position. If you don't want to be characters treated as single words, just delete the characters that you don't want.

The following example demonstrates the problem: the search pattern is 'Politik', so all strings are to be coded that start with `Politik`. But if a text like `Die Politologie (Politikwissenschaft) ...` occurs, then the string `Politikwissenschaft` will not be coded, because its enclosed in parentheses.

4.4.1 Rules for writing

In most cases the rules for typing texts with a type writer are okay. Hyphenation is to be avoided. No problems occur when there are dashes at the end of line, but errors – especially when generating a word list or its derivatives – occur if dashes are at the end of a line, e.g. *pre- and post-editing*.

It is also possible to separate characters from strings and treat these characters as strings, that is important when performing a coding with search patterns. Separation of characters is the defaults.

There are seven raw text formats available:

variable format using variable format control sequences within the text indicate the change of the values of external variables. Only the values of the external variables that change their values have to be specified. `KONTAKT.TXT` is a sample file.

fixed format Using fixed format all external variables have to be specified in *each* line. As long as the external variables have the same values, the text following the external variables belongs to the same text unit. `FEST.TXT` is a sample file.

line format Using line format every line is a text unit. The line counter is the only external variable. `ZEILE.TXT` is a sample file.

paragraph format Each paragraph is defined as a text unit. Paragraphs are separated by a blank line (CR/LF CR/LF). The paragraph counter is the only external variable. `ABSATZ.TXT` is a sample file.

page format Using page format each line is a text unit. There are two external variables: the page counter and the line counter. After x lines – this value can have a maximum value of 32767 – the first external variable is incremented. SEITE.TXT is a sample file.

SOLIS data bank Data from the SOLIS data bank of the Informationszentrums Sozialwissenschaften (Bonn) are processed. The external variables are predefined.

SRM data bank Data from the SRM data bank of the Erasmus-Universität Rotterdam are processed. The external variables are predefined.

4.4.2 Rules using variable format

The values of the external variables are specified with control sequences. These always start with a \$, and the values of the external variables of the *following* text unit are specified. Control sequences separate text units. The external variables are numbered in ascending order without gaps, starting with 1. The first control sequence at the beginning of the file of the raw text must contain **all external variables**. The following control sequences only have to contain the values of the external variables that change their values. If more than one external variable is changed, you must start the control sequence with the lowest one and specify the values of all others until the highest one. The values of the external variables are separated by dashes. If only one external variable is affected, the number of the external variable has to be specified after the \$.

The following pages show examples.

The rules that have to be followed while using external variables can be found in chapter 4.1 on page 42. The most important rules are:

- Up to 50 external variables are possible, one is absolutely necessary.
- The values of the external variables in the control sequences are separated by dashes.
- The maximum length of each external variable is 10 characters, the minimum length 1 character.
- Each external variable can consist of all characters except Tilde (~), number sign (#), dash (-) and the vertical bar (|, ASCII-value: 124). Blanks within the external variable are possible, case folding within the external variables is always disabled as well as the compression of multiple blanks. (p. 5 is not identical with p.5, TIME not with Time).

INTEXT does not change the values of the external variables.

1. example: coding open ended questions

1. control sequence: \$1(030295-1-1)

The external variables have the following values:

nr	variable	value
1	date	030295
2	number of person	1
3	number of question	1

The next control sequence only has to contain the values of the external variables that change their values. The control sequence for the next question is \$3(2). The values of the first two external variables do not change, the value of the third external variable is set to 2. Here is an example for profession, preferred television program and washing powder of three persons:

```
$1(130994-46-1) electrician
$3(2) Cross roads, Rich man poor man, Dallas
$3(3) Persil
$2(47-1) house wife
$3(2) Sesame street, Falcon Crest, Coronation street
$3(3) Ariel
$2(48-1) shop assistant
$3(2) Open university, Sky news, Match of the day
$3(3) Dash
```

2. example: analysis of printed media

There are two exmple for the analysis of printed media. This is the first example where only the necessary external variables are used.

1. control sequence: \$1(Time-030295-1)

The external variables have the following values:

nr	variable	value
1	medium	Time
2	data	030295
3	number of article	1

The next control sequence only has to contain the values of the external variables that change their values. The 154. article of "Newsweek" from 10th, November 1989 is defined by the following control sequence: \$1(Newsweek-101189-154).

The second example is taken from the dissertation of Claudia Kramer-Santel.

1. control sequence: \$1(Time-030295-culture-p. 3-headline)

The external variables have the following values:

nr	variable	value
1	medium	Time
2	date	030295
3	column	politics
4	page	p. 3
5	specialities	head line

The 4th external variable is the page number. For better readability no pure numerical solution was chosen, but a mixed one. This might cause problems during the statistical analysis, but it has the advantage that concordances and cross references are much easier to read. If a statistical analysis is planned, one has to have in mind that statistical software does have limitations in processing non-numerical data, e.g. SPSS only supports 8 characters in some procedures.

3. example: readability analysis

1. control sequence: \$1(gazette-1-1)

The external variables have the following values:

nr	variable	value
1	genre of text	newspaper
2	running number	1
3	sentence counter	1

The next control sequence only has to contain the values of the external variables that change their values. The control sequence for the next sentence is \$3(2). If the next text unit is the 3rd sentence of the 5th sample out of the genre prose, this is the control sequence: \$1(prose-5-3).

→The text unit must be the sentence.

4. example: literary research, e.g. style analysis

1. control sequence: \$1(Conrad-Nostromo-1-1)

The external variables have the following values:

nr	variable	value
1	author	Conrad
2	book	Nostromo
3	chapter counter	1
4	paragraph counter	1

The next control sequence only has to contain the values of the external variables that change their values. If the next unit is the 23rd paragraph of the 9th chapter of "Lord Jim" from the same author, the control sequence is:

\$2(Lord Jim-9-23).

5. example: news programs in television

1. control sequence: \$1(RTL-150486-1)

The external variables have the following values:

nr	variable	value
1	station	RTL
2	date	150486
3	item number	1

The next control sequence only has to contain the values of the external variables that change their values. If the next item of the same program follows, the control sequence is \$3(2). For the 4th item of RTL-news from 14th April, 1986, the control sequence is: \$1(RTL-140486-4).

Example with two news items (ARD Tagesschau from 14. April 1986):

\$1(ARD-140486-1) Last weekend 14 people were killed in severe race riots in South Africa. According to the police in Johannesburg 9 victims were blacks and killed because they were thought to cooperate with the government. \$3(2) 46 hindu pilgrims were killed in the north indian town Hatwar during a panic. While bathing in the holy river Ganges, some people fell, and a panic arose. The following crowd moved over them.

6. example: personal advertisements

1. control sequence: \$1(tip-020595-3-man-woman-self)

The external variables have the following values:

nr	variable	value
1	medium	tip
2	date	020595
3	running number	3
4	own gender	man
5	search gender	woman
6	type of image	self

The next control sequence only has to contain the values of the external variables that change their values. If the next text unit contains informations what peculiarities the woman shall have, the control sequence is: \$6(partner). If a woman looks for another woman and describes the type of relationship in the next unit, the control sequence is: \$3(4-woman-woman-relation), assuming that the ad is in the same medium on the same day. More examples are in the file KONTAKT.TXT.

\$1(160188-time-1-man-woman-self) Young man with a good job wants to meet a \$6(partner) woman between 30-40 years, also with children \$6(other) from the Cologne-Leverkusen-Gladbeck area \$6(relation) to build up a nice friendship. \$1(160188-time-2-man-woman-partner) Which young girl (up to 23 years) \$6(relation) is interested in conversation and spending days off with \$6(self) sensible and honest academic? \$6(other) answers with photos please \$1(160188-time-5-man-woman-self) Young man, 35 years, 176 cm tall, slim, with car, good income, looks for a \$6(partner) lovely and big busted woman for a \$6(relation) common future.

4.4.3 Rules for using fixed format

Using fixed format **each** line consists of two parts: the external variables at the beginning and after them the text. The part for the external variable contains all external variable in *each* line. For each external variable that column where it starts and its length in characters must be specified. The minimum length is 1 character, the maximum length is 10 characters. The external variables may overlap. The text must start in the same column each line, no external variables may follow.

Example for a raw text using fixed format:

```
1 1 1 Kleine als Grosse.
1 1 2 Die Kleine ist die Grosse, ist der Sieger
1 1 2 dieser Bundestagswahl.
1 1 3 Die FDP entstieg der Urne wie einem Jungbrunnen.
1 1 4 Die grob vereinfachte Wahlanalyse: CDU ja, Strauss
1 1 4 nein; Schmidt ja, Sozialismus nein, beides summierte
1 1 4 summierte sich unter Strich auf dem Konto der FDP.
```

The following rules have to be considered:

1. The line length of the raw text may not exceed 512 characters.
2. Each new text unit must begin on a new line.

4.4.4 Rules for using line format

The line format is useful for literary research just using a line number (1. external variable). Each line is a text unit, the line counter is incremented on each new line symbole (CR). One line may have up to 32500 characters. The line format allows the analysing of text without inserting control sequences. The name of the example file is ZEILEN.TXT.

4.4.5 Rules for using paragraph format

Using paragraph format each paragraph is a text unit. Paragraphs are separated by to end-of-line characters (CR/LF CR/LF). Only one external variable is supported, the paragraph counter. The name of the example file is ABSATZ.TXT.

4.4.6 Rules for using page format

Using page format each line is a text unit, the 1. external variable is the line counter like the line format. After x lines – this value must be specified by the user – the 2. external

variable is incremented by one. The name of the example file is SEITE.TXT.

4.4.7 Rules for processing the SOLIS data bank

The SOLIS data bank is provided by the Informationszentrum Sozialwissenschaften (Lennéstr. 30, 53113 Bonn, Germany) and consists of bibliographical data for the social sciences. The values of the external variables are fixed, the following conventions apply: the first one is always 1, the second is the running number within the databank, and the values of the third variable are specified as follows:

1. abstract
2. year
3. title
4. subtitle
5. author(s)
6. keywords

All other information that is part of the data bank is skipped. Parts of the data bank can be selected by defining a sample.

4.4.8 Rules for processing the SRM data bank

The external variables for processing the SRM data bank have the following meaning: the first variable is a code for the language:

1. english
2. german
3. dutch
4. french
5. italian
6. spanish

The second variable is the running number within the data bank, and the values of the third variable are specified as follows:

1. abstract
2. year
3. title
4. free
5. author(s)
6. keywords
7. language
8. type of entry (book, article)
9. magazine

All other information that is part of the data bank is skipped. Parts of the data bank can be selected, if you do not want to process all text units, you have to define a sample.

4.4.9 Parameters of the program

INTEXT/586 4.0 - 2/1997 - 15.05.1997 08:55	
program: ISYS	
application: generate system file	
	input/output files
name of raw text file	kontakt.txt
name of system file	kontakt.itx
	parameters
characters treated as words	.,;:!?()'"
format of the input file	free format
F1 help files F2 help parameters F3 change F4 start F10 end	

name of raw text file: the name the file that the raw text has. The name may contain drive and/or directory specifications.

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

format of the input file: the formats supported are free format, fixed format, line format, paragraph format, page format, and the databank formats SOLIS and SRM.

characters treated as words: Up to 30 different characters can be separated. These characters must be entered one after the other without blanks. Every character should be entered only once.

Start the program from MS-DOS level:: ISYS kontakt.txt kontakt.itx

4.4.10 Information messages

```
INTEXT/586 Version 4.0 - 2/1997 - 07.02.1997 16:17
  routine: ISYS
    application: generate system file
input file      kontakt.txt
output file     kontakt.itx
The following characters were separated .,;?!?()'"
- I 01:         914 lines read
- I 14:         202 different ID1 set absolute
- I 16:         358 different ID3 set absolute
- I 20:         94 strings in longest text unit
- I 21:         8829 strings processed
- I 22:         49404 characters processed
- I 23:         Wirtschaftswissenschaftler 26 characters line 403
- I 24:         560 text units written
- I 25:         15.766 strings/text unit
- I 26:         574 characters in longest text unit 507
- I 27:         204 characters separated at end of string
- I 28:         2078 characters separated at beginning of string
ISYS start:    16:17:54
ISYS end:      16:17:59
ISYS used 5 seconds CPU-time
```

4.4.11 Printed result of a system file with external variables

401	1	1	41	Junge , dynamische Frau (Akademikerin) , Anfang 30 , 176cm , gut aussehend , mit feminer Ausstrahlung , beruflich sehr zufrieden und in gesicherter Stellung , möchte nun etwas mehr Zeit für ihren privaten Lebensbereich aufwenden und sucht daher
401	1	2	14	einen niveaувollen , aufgeschlossenen und unternehmungslustigen Partner für gemeinsame Freizeitaktivitäten und eventuell mehr .
402	1	2	25	Männlicher Typ , möglichst gut aussehend , gut gebaut , gut situiert , intellektuell , originell , individuell , und anderes für repräsentative Zwecke von
402	1	1	8	ansonsten anspruchloser Sie (31) gesucht .
403	1	1	19	Vielseitig interessierte , naturverbundene , sportliche , schlanke , dynamische Sie , Akademikerin , 26/180 , CDU-Anhängerin , sucht
403	1	2	4	lieben adäquaten Ihn .
404	1	1	16	Selbstbewußte , attraktive Stierfrau (38) , mit Interesse an Kultur und Natur , sucht
404	1	2	5	offenen , zärtlichen Partner .
405	1	1	9	Alleinsein ist doof ! Attraktive , studierte Frau sucht
405	1	2	6	Mann mit Grips und Pep .
406	1	1	12	Witwe , 46/165 , kaufmännische Angestellte , ein erwachsener Sohn , sucht
406	1	2	6	zuverlässigen , etwa gleichaltrigen Lebenspartner .
406	1	1	3	Interessen vielseitig .
407	1	1	23	Bin Studentin , 21 , unabhängig in jeder Hinsicht , grün gesinnt , gläubig , ohne fromm zu sein , absolut zuverlässig und
407	1	2	13	auf der Suche nach einem lieben , treuen Freund bis 30 Jahre .
408	1	1	10	Gut aussehende , attraktive Sie , 25/180 , gutsituiert sucht
408	1	2	18	vorzeigbaren , attraktiven Ihn bis 35 , mit Herz , Charme und Verstand , für alltägliche Partnerschaft .
409	1	2	32	Hallo , Ihr mutigen Skorpione und Krebse ! Suche netten , blonden Ihn bis 35 zum Tanzen , Schmusen , Klönen und für alles , was zu zweit mehr Spaß macht .
409	1	1	24	Bin sympathische Fische-Frau , mittelblond , 32 , aufgeschlossen , tolerant und eigentlich ganz normal . Ganzfoto und Telefon erwünscht , arbeitslos zwecklos .
410	1	1	9	Ich , weiblich , 40/167 , sportlich , suche
410	1	2	6	unkomplizierten Mann bis 47 Jahre .
411	1	1	14	" Sie " , 41/160 , schlank , attraktiv , einfühlsam und zärtlich möchte
411	1	2	20	mit " Ihm " bis circa 48 Jahre (gern mit Schnäuzer) ab und zu den Alltag vergessen .
412	1	2	28	An einen netten , charaktervollen Akademiker (gern Lehrer , Mediziner . . .) : Vielleicht bist du selbstsicher , feinsinnig , aufgeschlossen und nachdenklich und möchtest
412	1	1	26	eine sensible , natürliche Studentin im Examen (27) mit christlicher Lebenseinstellung und vielseitigen sportlichen , musischen , kulturellen . . . Interessen kennenlernen ?
413	1	1	18	Sympathische blonde Frau , gute Figur , 35 Jahre , mit 9jähriger Tochter , sucht nach großer Enttäuschung
413	1	2	19	lieben Freund bis circa 40 Jahre . Er sollte zuverlässig sein , eine positive Lebenseinstellung und Durchsetzungsvermögen haben .
413	1	1	9	Ich hatte den Mut , diese Anzeige aufzusetzen ,
413	1	2	10	du solltest den Mut haben , mir zu antworten .
414	1	1	7	Ich möchte viel und gebe alles .
414	1	2	34	Wo sind eigentlich all die interessanten Männer , die man nicht an jeder Ecke trifft , welche noch frei oder wieder frei sind ? Vielleicht hatten wir bei irgendeiner Gelegenheit sogar schon Blickkontakt ?
414	1	3	7	Wir sollten es nicht dabei belassen !

5. Definition of samples

Many INTEXT-programs can not only work with the whole text, but also with parts of it – samples. At first you have to define the sample. For each external variable up to 10 limitations can be chosen, these are connected with a logical *or*, whereas within different external variables the combination is a logical *and*. An example shows what is meant: assume you want to select the newspaper "The Times", "Mirror", and "Daily Telegraph" and define these as a sample. During processing all text units are selected where the external variable "medium" has the values of the three papers (logical "or"). If you specify a date or a range of dates, only the text units out of the three papers are selected that are within the date ranges (logical "and"). The following examples show you how to define samples. The definitions are written to a file and can be used in the following analyses:

- word lists
- word sequences
- word permutations
- cross references
- content analysis
- personality structure analysis
- readability analysis
- concordances
- search units in text
- IRM (Intext Result Manager)

The external variables can be used to draw samples or to process the data in multiple steps. Details are described in chapter 4.4 on page 47 (program ISYS).

To define a sample you select menu project and submenu define sample. Four values are displayed, three can be altered.

- number of the external variable: the number of the external variable is requested. If you want to define the 5th external variable, you enter the value 5. Up to 10 limitations are possible for each external variable. If you want to finish, you enter a 0 to end the definition of the sample.
- running number of the external variable limitation is displayed and cannot be changed.
- minimum value: the smallest value that an external variable includes.
- maximum value: the maximum value that an external variable includes.

The following examples explain the definition of the samples.

1. example: coding of open ended questions, external variables are the date the number of the person and the number of the question. Only the questions 1, 3, and 5 are to be selected.

number of external variable:	3
minimum value:	1
maximum value:	1
number of external variable:	3
minimum value:	3
maximum value:	3
number of external variable:	3
minimum value:	5
maximum value:	5
number of external variable:	0

2. example: coding of open ended questions, external variables are the date number of person and number of question. The first three questions of the first 100 persons are to be selected.

number of external variable:	2
minimum value:	1
maximum value:	100
number of external variable:	3
minimum value:	1
maximum value:	3
number of external variable:	0

3. example: personal advertisements, external variable are medium, date, running number, own gender, searched gender, and type of image.

All partner images of women of the "Zeit" looking for men are to be selected.

number of external variable:	1
minimum value:	Zeit
maximum value:	Zeit
number of external variable:	4
minimum value:	woman
maximum value:	woman
number of external variable:	5
minimum value:	man
maximum value:	man
number of external variable:	6
minimum value:	partner
maximum value:	partner
number of external variable:	0

6. Analyses of texts

Analysing texts means that ones analyses their vocabulary. The term vocabulary is used with `INTEXT` for word lists, word sequences, word permutations, and cross references. These can be generated, reduced, and compared with each other. Vocabularies are useful for checking the spelling of the text, to describe the text, and as a basis for constructing a content analytical category system. Sometimes they are huge in size, and therefore they should be reduced.

The following criteria can be used to exclude strings from processing:

- external variables in form of a sample, see chapter 5 on page 59.
- length, measured in number of characters
- frequency, both absolute values or in per cent (%) or per mille (‰), e.g. 3,4 ‰. All values are inclusive. If e.g. the minimum length is 3 and the maximum length is 10, then all strings with at least 3 and at most 10 characters are processed.
- occurrence in a `STOP` word file, e.g. `ENGLISH.STP`. Then all entries of the `STOP` word file are not written to the vocabulary. The file `ENGLISH.STP` – which is part of `INTEXT` – contains punctuation marks, articles, pronouns, and prepositions. The entries in this file need not to be sorted by alphabet. Processing takes a lot more time then without `STOP` words.

All criteria can be combined. Length and frequency are specified by minimum and maximum values (inclusive values).

For each vocabulary one can process the whole text or a pre-defined sample, ignore differences due to case folding, control the format (normal or reverse), and the justification (left- or right justified).

The maximum file size that can be processed in one run is dependent on the available RAM and on the configuration of the operating system, using a sort order table needs a little more space:

$$\text{storage} = \text{size of file} * \text{TTR}$$

Due to the fact that the TTR becomes lower the more text you process, in most cases 8 MB installed RAM are enough. If there is no more RAM available, it is not possible to process all text units of a file at once in one run. Then it is necessary to draw samples from the file, generate a word list of each sample, copy these word lists into one file, sort it by alphabet, and then merge it to generate a word list of the whole file.

The TTR with word lists decreases with increasing amount of text, with 500 KB text it is about 0.15, with word sequences and cross references ca. 1. Word permutations need as 10 times as much RAM as the size of the system file.

6.1 Word lists

A word list is a table of all strings that occur within the system file (mostly words) and their frequency. It is sorted ascending by alphabet.

6.1.1 Normal form

INTEXT/586 4.0 - 4/1997 - 15.05.1997 15:03	
program: WORDBOOK	
application: word list	
input/output files	
name of system file	kontakt.itx
file of word list:	kontakt.wb
parameters	
process all text units	yes
case folding enabled	yes
format of vocabulary	not reverse
Justification of vocabulary	left justified
selection criteria	
minimum length 1	minimum frequency 1
maximum length 80	maximum frequency 100000
name of STOP-word file	no
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

A word list in normal contains all strings and their frequency of a text that must have the form of an INTEXT system file. The following parameters are available:

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

file of word list: the name of the file where the word list is written to. The name may contain drive and/or directory specifications.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed. Details are described in chapter 5 on page 59.

case folding enabled Letters can be treated as the same, if they are different only in their case (lower or upper case).

format of vocabulary: normal form or reverse form.

justification of vocabulary: left justified or right justified.

minimum length: the minimum number of characters a string must have to be included in the vocabulary.

maximum length: the maximum number of characters a string may have to be included in the vocabulary.

minimum frequency: the minimum number of occurrences a string must have to be included in the vocabulary.

maximum frequency: the maximum number of characters a string may have to be included in the vocabulary.

name of STOP-word file: If you enter a valid file name, all strings that are in the STOP-word file will not be processed.

whole words: If you answer with yes, the strings to be excluded must be identical with the STOP word, if you answer with no, the STOP-word can be any part of the string. Case folding is enabled. Example: if the STOP-word "men" is not defined as a whole word, then also the strings "women" and "amen" are excluded.

6.1.2 Information messages

INTEXT/586 Version 4.0 - 4/1997 - 15.05.1997 15:58

 routine: WORDBOOK
 application: word list
input file kontakt.itx
output file kontakt.wb
sort table ISM.DEF used
case folding enabled

statistics:

strings (token) read:

- I 01: 560 text units
- I 02: 6157 words
- I 03: 414 numbers
- I 04: 2258 other
- I 05: 8829 total of strings
- I 06: 10.995 words/text unit
- I 07: 0.739 numbers/text unit
- I 08: 4.032 other/text unit
- I 09: 15.766 total of strings/text unit
- I 10: 0 ID-errors

types token TTR type of string

strings written:

- I 21: 1968 6157 0.320 words
- I 22: 130 414 0.314 numbers
- I 23: 13 2258 0.006 other
- I 24: 2111 8829 0.239 total of strings

WORDBOOK started: 15:58:18

WORDBOOK ended : 15:58:28

WORDBOOK needed 10 seconds CPU-time

6.1.3 Reverse word list

A reverse word list contains all strings from right to the left. Also word sequences, word permutations, and cross references can be reversed.

INTEXT/586 4.0 - 4/1997 - 15.05.1997 15:03	
program: WORDBOOK	
application: reverse word list	
input/output files	
name of system file	kontakt.itx
file of word list:	kontakt.wbr
parameters	
process all text units	yes
case folding enabled	yes
format of vocabulary	reverse
justification of vocabulary	right justified
selection criteria	
minimum length 1	minimum frequency 1
maximum length 80	maximum frequency 100000
name of STOP-word file	no
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

The following parameters are available:

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

file of word list: the name of the file where the word list is written to. The name may contain drive and/or directory specifications.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed. Details are described in chapter 5 on page 59.

case folding enabled Letters can be treated as the same, if they are different only in their case (lower or upper case).

format of vocabulary: normal form or reverse form.

justification of vocabulary: left justified or right justified. The strings are formatted with leading blanks, so that the strings are right justified, that means all last letters are on the same position.

line length in characters Here the length of each string inclusive of leading blanks is requested. Values between 20 and 40 make sense. The ISYS program (see chapter 4.4 on page 47) displays in message I 25, how long the longest string is, and that must be the minimum value, otherwise the string will be truncated at the beginning of the word.

minimum length: the minimum number of characters a string must have to be included in the vocabulary.

maximum length: the maximum number of characters a string may have to be included in the vocabulary.

minimum frequency: the minimum number of occurrences a string must have to be included in the vocabulary.

maximum frequency: the maximum number of characters a string may have to be included in the vocabulary.

name of STOP-word file: If you enter a valid file name, all strings that are in the STOP-word file will not be processed.

whole words: If you answer with yes, the strings to be excluded must be identical with the STOP word, if you answer with no, the STOP-word can be any part of the string. Case folding is enabled. Example: if the STOP-word "men" is not defined as a whole word, then also the strings "women" and "amen" are excluded.

6.1.4 Printed results of a word list (normal form)

1	absehen	1	äußern	1	allgemeinen	1	Anfang
1	absetzbaren	3	äußerst	1	Alliierten	1	Anfangs
1	Absicht	4	äußerte	1	allzu	1	Anfassen
1	absichten	1	Äußerung	1	Alpenrepublik	1	Anflug
1	absolviert	1	Äußerungen	102	als	1	Anfrage
1	Absperrungen	2	AFG	21	Also	12	Angaben
2	Abstand	1	AG	6	alt	3	angeblich
1	Abstecher	1	Agenten	7	alte	1	angeblicher
2	Abstiegsrunde	1	Agententätigkeit	4	alten	3	Angebot
1	Abstimmung	1	Agraralkohol	1	altenglischen	2	angeboten
1	Absturz	1	Agrarfabriken	6	Alter	1	angedeutet
1	Abteilungen	1	Agrargipfel	1	Altern	3	angedroht
2	Abteilungsleiter	8	Agrarpolitik	94	am	9	angegriffen
1	Abu	2	Agrarpreise	1	amderen	1	angeheizt
1	abwartend	1	Agrarpreisverhand*	1	America	1	Angehörige
1	Abwehrfehler	3	Agression	8	Amerika	3	Angehörigen
1	Abwehrfront	1	Ahne	1	Amerikahaus	1	angehört
1	abzufangen	1	Akademie	46	Amerikaner	1	Angeklagte
1	abzuhalten	5	Akt	4	Amerikanerin	3	angeklagten
1	abzuholen	1	Aktenberge	9	Amerikanern	5	angekündigt
1	abzulehnen	30	Aktion	2	amerikanisch-lib*	1	angelegt
1	abzunehmen	5	Aktionen	77	amerikanische	1	angemeldeten
1	abzuschließen	1	aktiv	85	amerikanischen	3	angemessen
1	abzuwarten	1	Aktivität	8	amerikanischer	3	angenommen
1	abzuwickeln	10	Aktivitäten	1	Amerikareise	1	angerichtet
2	acht	3	aktuelle	3	Amis	3	angesagt
2	achtbares	1	aktuellen	1	Amokfahrer	1	angeschlagen
1	achten	1	akute	1	Amoklauf	1	angeschlossen
1	Achtung	1	Akzente	7	Amt	1	angesehen
1	Ackerbohnen	1	Al-Asia-Kaserne	1	Amtes	4	Angesichts
2	Action	1	Alarmbereitschaft	2	amtierende	1	angesprochen
2	ADAC	1	alarmiert	3	amtierenden	1	angesteckt
1	Adeltraut	1	Alarmstimmung	2	amtliche	1	Angestellten
2	Administration	1	Albrecht	1	Amtsantritt	1	Angestrebte
3	Adolf	1	Alex	4	Amtskollegen	1	Angewiesen
3	Adoptivtochter	2	Alexander	92	an	1	angeworben
1	Adresse	1	Algeriens	1	Anbaus	3	Angreifer
5	Ägypten	1	Ali	1	Anbetracht	2	angreift
2	Ägyptens	1	Aliierten	1	Anbiatern	52	Angriff
1	Ägypter	2	all	19	andere	8	Angriffe
1	ägyptische	2	Allah	4	anderem	7	Angriffen
1	ähnlich	26	alle	14	anderen	2	Angriffes
1	Ähnlich	11	allein	2	anderer	7	Angriffs
1	ältere	1	Alleingang	3	andererseits	1	Angriffsentscheidung
1	älteren	27	allem	3	anderes	1	Angriffsplan
2	Ämter	9	allen	1	Anderlecht	1	Angriffswelle
1	ändert	2	aller	5	anders	2	Angriffsziele

6.1.5 Printed result of a reverse word list

4	(1	netraG	1	niloP
4)	1	tiehnegrobeG	3	muaR
44	,	1	gitsieg	1	eger
14	.	1	hciltnegeleg	1	mellovsthciskcür
1	810	2	ehciltnegeleg	4	knalhcs
1	mc471	1	emasniemeg	1	enöhcs
1	mc571	1	enreg	1	bierhcs
1	mc671	1	ehcärpseG	1	grebeeS
1	j32	1	dnehessuatug	2	rhes
1	52	1	etug	1	nerehcistsbles
1	j62	1	metug	3	eiS
1	j04-03	1	netug	1	hciltrops
1	j53	1	setug	1	ehcus
1	j54	1	raaH	1	nehcus
1	j05	1	nehcilkrewdnah	1	tshcus
1	j55	1	suaH	5	thcus
1	j26	1	uarfsuaH	1	sfferT
1	j56	1	ehcilsuäh	1	neuert
1	gk07	1	sinredniH	6	dnu
2	?	1	hcl	1	tewtiwrev
1	rekimedakA	1	mi	1	lativ
1	na	1	netnegilletni	1	egisublov
1	hcua	1	nereisseretni	1	knalhcslllov
1	neuabfua	1	erhaJ	1	eknalhcslllov
1	sua	1	egnuj	1	ehcilbiew
1	egülfuA	2	regnuJ	1	sehcleW
1	fureB	1	hcsilohtak	1	rewtiW
1	gnuuerteB	1	niek	1	ehcilträz
1	netfirhcsuzdliB	1	nenrelnennek	1	tiekhcilträZ
1	sib	1	dniK	1	tfnukuZ
1	tsiB	1	redniK	1	skcewz
1	dnolb	1	nredniK	1	nehcsiwz
2	emaD	1	kcebdalG-nesukreveL-nlök		
1	med	2	ebeil		
1	ned	2	nebeil		
1	ud	1	netsebsnebeil		
1	ehE	1	ledäM		
1	nnamehE	2	nnaM		
2	nehcilorhe	1	hcsubreeM		
1	enie	1	rhem		
2	nenie	1	nehcsneM		
1	renie	1	rim		
1	ehcafniE	5	tim		
1	nemaslhüfnie	1	metlalettim		
1	nemmmokniE	1	segnallettim		
1	rE	2	ethcöm		
1	ehcsitore	1	etten		
1	otoF	1	ierfnitokin		
1	uarF	1	redo		
1	gnutlatsegtiezierF	2	rentraP		
2	tfahcsdnuerF	2	nirentraP		
3	rüf	1	negawtfarknenosreP		

6.2 Word sequences

An analysis technique that exceeds the limits of single words is the generation of word combinations. These are parts of a text unit that consist of x words, the value of x is to be defined. If it is 1, a word list is generated.

If a text unit is This is a test and word sequences with 2 words are to be generated, the output (unsorted) looks like this:

This is
is a
a test

6.2.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 15.05.1997 09:01	
program: WORDBOOK	
application: word sequence	
14935 KB RAM free	input/output files
name of system file	kontakt.itx
file name of word sequences	kontakt.wbc
parameters	
process all text units	yes
case folding enabled	yes
format of vocabulary	not reverse
justification of vocabulary	left justified
sort criterion for word sequences	sorted by first string
number of strings	2
selection criteria	
minimum length 1	minimum frequency 1
maximum length 80	maximum frequency 100000
name of STOP-word file	no
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

file name of word sequences: the name of the file where the word sequences are written to. The name may contain drive and/or directory specifications.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed. Details are described in chapter 5 on page 59.

case folding enabled Letters can be treated as the same, if they are different only in their case (lower or upper case).

format of vocabulary: normal form or reverse form.

justification of vocabulary: left justified or right justified.

sort criterion for word sequences: the word sequences can be sorted alphabetically by the first string of the sequences or by the last string.

number of strings: This value defines, how many words form a word sequence. The default value is 2, the highest is the number of words in the shortest text unit. For example, if the shortest text unit consists of 9 words, the highest value that makes sense is 9.

minimum length: the minimum number of characters a string must have to be included in the vocabulary.

maximum length: the maximum number of characters a string may have to be included in the vocabulary.

minimum frequency: the minimum number of occurrences a string must have to be included in the vocabulary.

maximum frequency: the maximum number of characters a string may have to be included in the vocabulary.

name of STOP-word file: If you enter a valid file name, all strings that are in the STOP-word file will not be processed.

whole words: If you answer with yes, the strings to be excluded must be identical with the STOP word, if you answer with no, the STOP-word can be any part of the string. Case folding is enabled. Example: if the STOP-word "men" is not defined as a whole word, then also the strings "women" and "amen" are excluded.

6.2.2 Information messages

```

INTEXT/PC 4.0 - 4/1997 - 28.04.1997 06:03
      routine: WORDBOOK
      application: word combinations
input file      kontakt.itx
output file     kontakt.wbc
sort oder table ISM.DEF used
case folding enabled
sorted by last word
range: 3 strings
Statistics:
Strings (token) read:
- I 01:      560 text units
- I 02:      5493 words
- I 03:      399 digits
- I 04:      1819 other
- I 05:      7711 strings in total
- I 06:      9.809 words/text unit
- I 07:      0.712 digits/text unit
- I 08:      3.248 other/text unit
- I 09:      13.770 strings in total/text unit
Types      Token   TTR   type of string
strings written:

- I 21:      5241      5493   0.954 words
- I 22:      336       399    0.842 digits
- I 23:      1416      1819   0.778 other
- I 24:      6993      7711   0.907 strings in total
WORDBOOK start: 06:03:41
WORDBOOK end:   06:03:48
WORDBOOK needed 7 seconds CPU-time

```

6.2.3 Printed results of word sequences

1	attraktiv (40	1	Auch in uns	1	auf meinen ersten
1	attraktiv , einfühlsam	1	auch jünger ,	1	auf Partnerschaft ,
1	attraktiv , mit	1	auch jünger .	1	auf Treue und
1	attraktiv , spontan	1	auch keine "	1	auf unserer Terrasse
1	attraktiv , sucht	1	auch keine Prinzessin	1	Aufbau einer harmonischen
1	attraktiv so sagt	1	auch keine solche	1	aufbauen . Raum
1	attraktive , etwas	1	auch künstlerisch aktiven	1	Aufgabe von Dauer
1	attraktive , kinderlose	1	auch Löwe ,	1	aufgeschlossen , sucht
1	Attraktive , studierte	1	auch mal auf	1	aufgeschlossen , tolerant
1	attraktive Erscheinung ,	1	auch mal versuchen	1	aufgeschlossen und nachdenklich
1	Attraktive Frau ,	1	auch meist ,	1	aufgeschlossen und nicht
1	attraktive Partnerin ,	2	auch mit Kind	1	aufgeschlossen und vorzeigbar
1	Attraktive Sie ,	1	auch nicht von	1	aufgeschlossene , ungebundene
1	attraktive Sie zur	1	auch oft neugierig	1	aufgeschlossene Partnerin ,
1	Attraktive Skorpionin ,	1	auch schon .	1	aufgeschlossenen , lebensfrohen
1	attraktive Stierfrau (1	auch schöner 20jähriger	1	aufgeschlossenen , liebenswerten
1	attraktive und erotische	1	auch so ?	1	aufgeschlossenen und lustigen
1	Attraktive und fröhliche	1	auch über die	1	aufgeschlossenes , entdeckungsfreudiges
1	Attraktive Vierzigerin ,	1	auch von Deinem	1	aufgeschlossenes , lebenslustiges
1	attraktive weibliche Wesen	1	auch vor hat	1	aufgewachsen , in
1	attraktiven , schlanken	1	auch Witwe mit	1	aufheitert , überrascht
1	attraktiven Ihn bis	1	auch zur Liebe	1	aufrichtig , in
1	Attraktiver , selbstbewußter	1	auf , schreib	1	aufrichtige , adäquate
1	Attraktiver Student hat	1	auf abwechslungsreiche Zeiten	1	aufrichtige Partnerin (
1	attraktive Sie ,	1	auf alles ,	1	aufrichtigen Ihn bis
1	Attributen ausgestattet ,	1	auf bloße oberflächliche	1	aufwenden und sucht
1	auch " nur	1	auf deine Zuschrift	1	Augen , ernst
1	auch albern)	1	auf Deiner Liane	1	Augen , manchmal
1	auch alleine erziehen	1	auf dem Besen	1	Augen , schlank
1	auch als Aufgabe	1	auf den Arm	1	Augen-Blicken noch zu
1	auch anders ,	1	auf der Erde	1	August-Strindberg-Preises " ,
1	auch Angst ,	1	auf der Suche	1	aus , mittelblond
1	Auch auf die	1	auf der Welt	1	aus 20jähriger Ehe
1	auch Ausländer)	1	auf die erde	2	aus dem Raum
1	auch Ausländerin angenehm	1	auf die Gefahr	1	aus dem Weg
1	auch Cabriofahren ,	1	auf diese Anzeige	1	aus guter Familie
1	auch Camping ?	1	auf diese Weise	1	aus Parität gesicherten
1	auch das Wieder-Alleinsein	1	auf diesem ungewöhnlichen	1	aus Selbstzweifeln und
1	auch Dein Leben	1	auf diesem Weg	1	ausgeglichenes Wesen ,
1	auch dem Leben	4	auf diesem Wege	1	ausgeprägter Persönlichkeit und
1	auch die Ruhe	1	auf ein zufälliges	1	ausgeschlossen . Chiffre
1	auch die Wärme	1	auf eine gemeinsame	1	ausgestattet , wünscht
1	auch Du mir	1	auf einen männlichen	1	ausgestattet glaubt .
1	auch ein- bis	1	auf erotische Nähe	1	aushecken , radfahren
1	auch etwas kräftige	1	auf freundliche und	1	Ausland (gute
1	auch Fehler haben	1	auf Freundschaft mit	1	Ausland verbringt .
1	auch finanziell (1	auf Händen trägt	1	Ausländer) denen
1	auch finanziell ,	1	auf ihn .	1	Ausländerin angenehm .
1	auch fühlen kann	1	auf Kinder ,	1	Auslandsleben liebt ,
1	auch für möglich	1	auf konventionelle Beziehung	1	ausleben kann .
1	auch für politische	1	auf Leistungssport fixiert	1	aussehen und lustig
1	auch gerne nach	1	auf loser Zunge	1	aussehend , 179

6.3 Word permutations

Word permutations are performed for each text unit. They consist of two word combinations: the first word with the second and all other following words, the second word with the third word and all other following words, and so on. If a text unit is `This is a test`, the output (unsorted) looks like this:

```
This is
This a
This test
is a
is test
a test
```

Because the word permutations are sorted internally, the order of the word permutations is different in the output file. Word permutations need a lot of RAM, depending of the length of the text units and the size of the system file.

6.3.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 15.05.1997 09:02	
program: WORDBOOK	
application: word permutations	
14935 KB RAM free	input/output files
name of system file	kontakt.itx
file name of word permutations	kontakt.wbp
parameters	
process all text units	yes
case folding enabled	yes
format of vocabulary	not reverse
justification of vocabulary	left justified
selection criteria	
minimum length 1	minimum frequency 1
maximum length 80	maximum frequency 100000
name of STOP-word file	no
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

file name of word sequences: the name of the file where the word sequences are written to. The name may contain drive and/or directory specifications.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed. Details are described in chapter 5 on page 59.

case folding enabled Letters can be treated as the same, if they are different only in their case (lower or upper case).

format of vocabulary: normal form or reverse form.

justification of vocabulary: left justified or right justified.

minimum length: the minimum number of characters a string must have to be included in the vocabulary.

maximum length: the maximum number of characters a string may have to be included in the vocabulary.

minimum frequency: the minimum number of occurrences a string must have to be included in the vocabulary.

maximum frequency: the maximum number of characters a string may have to be included in the vocabulary.

name of STOP-word file: If you enter a valid file name, all strings that are in the STOP-word file will not be processed.

whole words: If you answer with yes, the strings to be excluded must be identical with the STOP word, if you answer with no, the STOP-word can be any part of the string. Case folding is enabled. Example: if the STOP-word "men" is not defined as a whole word, then also the strings "women" and "amen" are excluded.

6.3.2 Information messages

```

INTEXT/586 4.0 - 4/1997 - 28.04.1997 06:05
    routine: WORDBOOK
    application: word permutations
input file      kontakt.itx
output file     kontakt.wbp
sort oder table ISM.DEF used
case folding enabled
Statistics:
Strings (token) read:
- I 01:         560 text units
- I 02:         5493 words
- I 03:          399 digits
- I 04:         1819 other
- I 05:         7711 strings in total
- I 06:         9.809 words/text unit
- I 07:         0.712 digits/text unit
- I 08:         3.248 other/text unit
- I 09:        13.770 strings in total/text unit
      Types      Token   TTR   Type of string
strings written:
- I 21:         5241      5493   0.954 words
- I 22:          336      399   0.842 digits
- I 23:         1416      1819   0.778 other
- I 24:         6993      7711   0.907 strings in total
WORDBOOK start: 06:05:41
WORDBOOK end:   06:05:48
WORDBOOK needed 7 seconds CPU-time

```

6.3.3 Printed results of word permutations

1	a	2	ab für	1	Abende und
1	a virgin	1	ab gutmütig	1	Abende undsoweiter
1	Aachen-Köln-Düsseldorf- aber	1	ab Hessen	1	Abende wie
1	Aachen-Köln-Düsseldorf- Aktivität	1	ab hohe	5	aber
1	Aachen-Köln-Düsseldorf- anderen	1	ab humorvoll	1	aber Abend
1	Aachen-Köln-Düsseldorf- anderswo	1	ab ideenreich	2	aber aber
1	Aachen-Köln-Düsseldorf- Arbeit	1	ab in	1	aber allein
1	Aachen-Köln-Düsseldorf- bei	1	ab Jahre	1	aber als
1	Aachen-Köln-Düsseldorf- beim	1	ab jovial	1	aber alten
1	Aachen-Köln-Düsseldorf- Bereiche	1	ab keck	1	aber am
1	Aachen-Köln-Düsseldorf- denen	1	ab kennenlernen	1	aber an
1	Aachen-Köln-Düsseldorf- der	1	ab lustig	1	aber anderen
1	Aachen-Köln-Düsseldorf- eine	1	ab manierlich	1	aber Anfang
1	Aachen-Köln-Düsseldorf- einer	1	ab Menschlichkeit	1	aber Anlehnung
1	Aachen-Köln-Düsseldorf- es	1	ab mit	2	aber auch
1	Aachen-Köln-Düsseldorf- Frauen	1	ab muß	3	aber auf
1	Aachen-Köln-Düsseldorf- gibt	1	ab nachdenklich	1	aber aus
1	Aachen-Köln-Düsseldorf- hätte	1	ab nicht	1	aber außer
1	Aachen-Köln-Düsseldorf- ich	1	ab Niveau	1	aber Bedingung
1	Aachen-Köln-Düsseldorf- in	1	ab oder	1	aber Bedürfnis
1	Aachen-Köln-Düsseldorf- kennengelernt	1	ab offensiv	1	aber Bereiche
1	Aachen-Köln-Düsseldorf- Lieber	1	ab peppig	1	aber beruflich
1	Aachen-Köln-Düsseldorf- noch	1	ab Persönlichkeit	1	aber berufstätig
1	Aachen-Köln-Düsseldorf- oder	1	ab quietschfidel	2	aber Beziehung
1	Aachen-Köln-Düsseldorf- Partnerin	1	ab reichen	1	aber Bild
1	Aachen-Köln-Düsseldorf- sind	1	ab romantisch	1	aber bitte
1	Aachen-Köln-Düsseldorf- stark	1	ab salonfähig	1	aber blond
1	Aachen-Köln-Düsseldorf- Studium	1	ab schon	1	aber Camping
1	Aachen-Köln-Düsseldorf- und	1	ab sein	1	aber Charmante
1	Aachen-Köln-Düsseldorf- unterrepräsentiert	1	ab treu	1	aber Chiffre
1	Aachen-Köln-Düsseldorf- zuviele	1	ab um	1	aber da
1	ab alles	1	ab umsichtig	1	aber dafür
1	ab Alltag	2	ab und	2	aber das
1	ab anderswo	1	ab vergessen	1	aber daß
1	ab Anmutig	1	ab verschwenderisch	1	aber dem
1	ab auch	1	ab von	2	aber den
1	ab ausgeprägt	1	ab widerspenstig	1	aber denen
1	ab beheimatende	1	ab Zentimeter	1	aber der
1	ab bemüht	1	ab zu	1	aber dichte
1	ab bezaubernd	1	Abend am	1	aber Disko-Illusionen
1	ab Bindung	1	Abend auf	2	aber Diskretion
1	ab Charakterzüge	2	Abend im	1	aber doch
1	ab charmant	1	Abend Kaminfeuer	1	aber dominanter
1	ab damenhaft	1	Abend Sommer	1	aber echter
1	ab den	1	Abend Terrasse	2	aber eine
1	ab Du	1	Abend unserer	1	aber einem
1	ab ehrliche	1	Abend Winter	2	aber einen
2	ab eine	1	Abend zu	1	aber einer
1	ab Einige	1	Abend zweit	1	aber Einseitigkeit

6.4 Cross references

A cross references of a text consist of all occurrences of each string together with its external variables and the position of the string (number of the string in the text unit), sorted by alphabet.

In system with hierarchical external variables cross references should be unique, that means, that no string should have the same external variables and the same position. If this is the case, there maybe incorrect external variables in the text.

Samples of the text units can be drawn. Words, digits and other types of strings are counted, and the average length of a text unit (in words, not in strings) is computed. If the sort order table ISM.DEF exists, it will be used (see page 122 for details). Also case folding can be enabled or disabled. The frequency of the string is written after the last reference into a separate line. Strings can be excluded from processing if they occur in a STOP wordlist.

6.4.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 15.05.1997 09:08	
program: WORDBOOK	
application: cross references	
14935 KB RAM free	input/output files
name of system file	kontakt.itx
file of cross references	kontakt.xrf
parameters	
process all text units	yes
case folding enabled	yes
format of vocabulary	not reverse
justification of vocabulary	left justified
number of references per line	3
selection criteria	
minimum length 1	minimum frequency 1
maximum length 80	maximum frequency 100000
name of STOP-word file	no
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

file name of cross references: the name of the file where the cross references are written to. The name may contain drive and/or directory specifications.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed. Details are described in chapter 5 on page 59.

case folding enabled Letters can be treated as the same, if they are different only in their case (lower or upper case).

format of vocabulary: normal form or reverse form.

justification of vocabulary: left justified or right justified.

number of references per line: Here the number of cross references per line are to be specified. The minimum value is 1, every cross reference starts with a new line. It consists of the external variables separated with tildes. The cross references are separated by a blank.

minimum length: the minimum number of characters a string must have to be included in the vocabulary.

maximum length: the maximum number of characters a string may have to be included in the vocabulary.

name of STOP-word file: If you enter a valid file name, all strings that are in the STOP-word file will not be processed.

whole words: If you answer with yes, the strings to be excluded must be identical with the STOP word, if you answer with no, the STOP-word can be any part of the string. Case folding is enabled. Example: if the STOP-word "men" is not defined as a whole word, then also the strings "women" and "amen" are excluded.

Start the program from MS-DOS level:

```
wb kontakt.itx kontakt.xrf -c -t -xrf
```

6.4.2 Information messages

INTEXT/586 Version 4.0 - 4/1997 - 26.05.1997 20:30

routine: WORDBOOK

application: cross references

input file kontakt.itx

output file kontakt.xrf

case folding enabled

statistics:

strings (token) read:

- I 01: 560 text units
- I 02: 6157 words
- I 03: 414 numbers
- I 04: 2258 other
- I 05: 8829 total of strings
- I 06: 10.995 words/line
- I 07: 0.739 numbers/line
- I 08: 4.032 other/line
- I 09: 15.766 total of strings/line
- I 10: 15 ID-errors

types	token	TTR	type of string
-------	-------	-----	----------------

strings written:

- I 21:	6153	6157	0.999 words
- I 22:	414	414	1.000 numbers
- I 23:	2247	2258	0.995 other
- I 24:	8814	8829	0.998 total of strings

WORDBOOK started: 20:30:10

WORDBOOK ended : 20:30:51

WORDBOOK needed 41 seconds CPU-time

6.4.3 Printed results of cross references

Akademiker

160188~Zeit~2~Mann~Frau~Selbst~5

an

160188~Zeit~2~Mann~Frau~sonstiges~2

auch

160188~tip~69~Frau~Frau~Beziehung~3

aufbauen

160188~Zeit~1~Mann~Frau~Beziehung~4

aus

160188~Zeit~1~Mann~Frau~sonstiges~1

Ausflüge

160188~Zeit~2~Mann~Frau~Beziehung~5

Beruf

160188~Zeit~1~Mann~Frau~Selbst~6

Betreuung

160188~Zeit~6~Mann~Frau~Fremd~3

Bildzuschriften

160188~Zeit~2~Mann~Frau~sonstiges~1

bis

160188~Zeit~2~Mann~Frau~Fremd~5

Bist

160188~Zeit~6~Mann~Frau~Fremd~5

blond

160188~Zeit~4~Frau~Mann~Selbst~7

Dame

160188~Zeit~4~Frau~Mann~Selbst~1 160188~Zeit~9~Frau~Mann~Selbst~2

dem

160188~Zeit~1~Mann~Frau~sonstiges~2

den

160188~Zeit~4~Frau~Mann~Fremd~1

du

160188~Zeit~6~Mann~Frau~Fremd~6

Ehe

160188~Zeit~6~Mann~Frau~Fremd~14

Ehemann

160188~Zeit~7~Frau~Mann~Fremd~6

ehrlichen

160188~Zeit~7~Frau~Mann~Fremd~3 160188~Zeit~8~Frau~Mann~Fremd~7

eine

160188~Zeit~1~Mann~Frau~Beziehung~1

einen

160188~Zeit~7~Frau~Mann~Fremd~2 160188~Zeit~8~Frau~Mann~Selbst~8

einer

160188~Zeit~1~Mann~Frau~Selbst~9

Einfache

160188~Zeit~9~Frau~Mann~Selbst~1

emfühltsamen

160188~Zeit~2~Mann~Frau~Selbst~1

Einkommen

160188~Zeit~5~Mann~Frau~Selbst~14

6.5 Comparison of vocabularies

The WORDCOMP program compares two vocabularies in four types of analyses:

- Comparison of two vocabularies with the differences of the strings
- Output of the strings, that only occur in the second vocabulary but not in the first one
- statistics only, the comparison of the vocabularies is suppressed
- Comparison of two vocabularies of the strings that occur in both vocabularies

Statistics include the number of words of each vocabulary, the number of words and their frequency are written to the output file. Both vocabularies **must** be sorted with the same sort order table. The strings are divided into words, numbers and others, also the type-token-ratio (TTR) is computed. Also the number of exclusive strings which occur only in one word list and the number of inclusive strings that occur in both vocabularies and the appropriate TTR-values are computed.

The complete comparison can be written to the output file in three ways:

- short format: output are the frequencies of the first file, the second file, the differences between the two frequencies and the string. The frequencies are formatted in 9 digits. If a string occurs only in one file, the frequency field of the other file is left blank, the difference is not computed.

column	contents
1 - 9	frequency of the word in the 1. file
10 - 18	frequency of the word in the 2. file
19 - 27	difference of the frequencies
28	free
29 -	word

- long format: the first 39 characters of the strings of each file followed by its frequency displayed in 7 digits. Between the two columns the differences of the frequencies are shown in 7 digits.

columns	contents
1 - 7	frequency of the word in the 1. file
8 - 46	word in the 1. file
47 - 53	difference of the frequencies
54 - 60	frequency of the word in the 2. file
61 - 99	word in the 2. file

- difference format: the frequencies of all strings that occur in both file are output as well as their difference and the string. The frequencies and the differences use 9 digits.

columns	contents
1 - 9	frequency of the word in the 1. file
10 - 18	frequency of the word in the 2. file
19 - 27	difference of the frequencies
28	free
29 -	word (unlimited length)

The short format is not as space consuming as the long format, the short format can be printed using more characters densities (10 or 12 cpi). The frequencies are 9 digits instead of 7 digits in the long format, and the string can have an unlimited length instead of being truncated after the 39. character without warning.

The difference format fits best for analyses of strings that are in both files. Strings longer 39 characters are not truncated like in short format.

6.5.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 15.05.1997		09:15
program: WORDCOMP		
application: complete vocabulary comparision		
input/output files		
name of 1. vocabulary file:		kontakt.wb
name of 2. vocabulary file:		nd.wb
parameters		
type of vocabulary comparision:		complete
output file of vocabulary comparision:		kontakt.wcp
format of the vocabulary comparision:		short format
F1	help files	F2 help parameters
F3	change	F4 start
F10	end	

name of 1. vocabulary file: the name of the file where the first vocabulary is stored. The name may contain drive and/or directory specifications.

name of 2. vocabulary file: the name of the file where the second vocabulary is stored. The name may contain drive and/or directory specifications.

type of vocabulary comparision: complete comparision: comparison of two vocabularies with the differences of the strings. new strings: Output of the strings, that only occur in the second vocabulary but not in the first one. statistics only: the

comparison of the vocabularies is suppressed. Comparison of two vocabularies of the strings that occur in both vocabularies

output file of vocabulary comparison: The file name of the results is specified here: either containing the complete comparison or the new strings.

format of the vocabulary comparison: The available formats are the long format and the short format.

Start the program from MS-DOS level:

```
wordcomp wb1.wb wb2.wb
```

6.5.2 Information messages

```
INTEXT/586 Version 4.0 - 4/1997 - 26.05.1997 14:16
```

```
routine: WORDCOMP
```

```
application: vocabulary comparison
```

```
input file 1 kontakt.wb
```

```
input file 2 kontest.wb
```

```
output file new.wb
```

```
Statistics:
```

	File 1			File 2			File 1 + file 2		
	Types	Token	TTR	Types	Token	TTR	Types	Token	TTR
strings read									
words	1963	6096	0.322	208	302	0.689			
digits	130	414	0.314	17	20	0.850			
other	26	2285	0.011	9	121	0.074			
sum	2119	8795	0.241	234	443	0.528			
exclusive strings:									
words	1755	3692	0.475	0	0	0.000			
digits	113	312	0.362	0	0	0.000			
other	17	46	0.370	0	0	0.000			
sum	1885	4050	0.465	0	0	0.000			
sum of common strings:									
words	208	2404	0.087	208	302	0.689	208	2706	0.077
digits	17	102	0.167	17	20	0.850	17	122	0.139
other	9	2239	0.004	9	121	0.074	9	2360	0.004
sum	234	4745	0.049	234	443	0.528	234	5188	0.045
WORDCOMP start:	14:16:37								
WORDCOMP end:	14:16:57								
WORDCOMP needed	20 seconds CPU-time								

6.5.3 Printed results of vocabulary comparisons

Figure 3: Vocabulary comparison in long format

				1	A-dream-comes-true
1	Aachen				
1	Aachen/Strasbourg				
1	Aachener				
86	ab	38	48	ab	
				1	ab170cm
				3	ab175cm
				3	ab180cm
				4	ab18J
				2	ab20J
				2	ab25J
				5	ab30j
				2	ab35J
				4	ab40j
				1	ab45J
				1	ab55J
				1	ab60j
				1	ab65J
9	Abb				
11	Abbau				
2	abbauen				
1	Abbaus				
1	Abberufung				
1	Abberufungen				
1	Abbild				
2	Abbildungen				
1	abbringen				
1	Abbruch				
1	Abdallah				
1	Abdeckung				
1	Abderrachid				
11	Abeba				
9	Abend	8	1	Abend	
1	Abendblattes				
1	Abende				
				1	Abenden
				2	Abendkleid
				1	Abendkleider
1	abendlichen				
1	abends	-2	3	abends	
2	Abendsingen				
1	Abendstunden				
				14	Abenteuer
				1	abenteuerliche
				1	abenteuerlustig
				1	Abenteuertypen
				9	Abenteurer
114	aber	23	91	aber	
				1	Aberglauben
1	aberkannt				

Figure 4: Vocabulary comparison in short format

	1		A-dream-comes-true
1			Aachen
1			Aachen/Strasbourg
1			Aachener
86	48	38	ab
	1		ab170cm
	3		ab175cm
	3		ab180cm
	4		ab18J
	2		ab20J
	2		ab25J
	5		ab30j
	2		ab35J
	4		ab40j
	1		ab45J
	1		ab55J
	1		ab60j
	1		ab65J
9			Abb
11			Abbau
2			abbauen
1			Abbaus
1			Abberufung
1			Abberufungen
1			Abbild
2			Abbildungen
1			abbringen
1			Abbruch
1			Abdallah
1			Abdeckung
1			Abderrachid
11			Abeba
9	1	8	Abend
1			Abendblattes
1			Abende
	1		Abenden
	2		Abendkleid
	1		Abendkleider
1			abendlichen
1	3	-2	abends
2			Abendsingen
1			Abendstunden
	14		Abenteuer
	1		abenteuerliche
	1		abenteuerlustig
	1		Abenteuertypen
	9		Abenteurer
114	91	23	aber
	1		Aberglauben
1			aberkannt

Figure 5: Vocabulary comparison in difference format

26	12	14	A
86	48	38	ab
9	1	8	Abend
1	3	-2	abends
114	91	23	aber
2	5	-3	abgesichert
1	3	-2	Abitur
4	2	2	absolut
5	2	3	absolute
5	1	4	Absturz
5	1	4	Abteilungsleiter
4	1	3	achten
14	6	8	Achtung
1	2	-1	Adam
2	12	-10	adäquate
2	7	-5	adäquaten
2	5	-3	Adresse
16	1	15	Afrika
2	3	-1	ähnlich
2	4	-2	ähnlichen
1	64	-63	Akademiker
1	2	-1	akademischen
13	1	12	Aktionen
23	13	10	aktiv
19	3	16	aktive
15	6	9	aktiven
4	1	3	Aktivität
26	3	23	Aktivitäten
2	5	-3	akzeptiert
188	18	170	Alle
32	43	-11	Allein
1	14	-13	Alleinstehende
97	18	79	allem
140	7	133	allen
127	3	124	aller
22	3	19	Allerdings
43	56	-13	alles
3	1	2	allgemein
2	4	-2	Allgemeinbildung
9	1	8	allgemeine
2	6	-4	Alltag
1	2	-1	alltäglichen
1	2	-1	Alltags
1	2	-1	Alpen
662	51	611	als
22	8	14	also
4	17	-13	alt
12	2	10	Alte

6.6 Searching within a vocabulary

One can search within vocabularies, it is faster than searching in system files.

Searchable are strings, depending on the type of vocabulary strings are words or parts of it. Up to 2 words can be search in word permutations, in word sequences this is dependent on the number of words. The search patterns are called GO-words, and one can use < and > showing that before a < or after a > characters are allowed (see word root chains, chapter 6.8 on page 94). Also ? and * can be used as wild card symbols. ? can be used more than once, * only once (similar to the usage in MS-DOS file names).

The following examples show which search patterns find which strings within a vocabulary:

search pattern	vocabulary string
man	man
<man	man, woman
<man>	Amanda

Using interactive entry of search patterns these are terminated by pressing RETURN. Two times RETURN means that all search patterns are specified.

GO-words can have two formats, the detection is done automatically:

1. as a file of search patterns with code and parameter field
2. each GO-word occupies a separate line in a file

6.6.1 Parameters of the program

INTEXT/586 4.0 - 2/1997 - 06.02.1997 14:15				
routine: WOBANA				
application: searching with GO-words				
input/output files				
file of vocabulary				kontakt.wb
vocabulary only GO words				kontakt.wbg
file of GO words				interactive entry
F1	help files	F2	help parameter	F3
				change
				F4
				start
				F10
				end

file of vocabulary: the file that contains the text to be searched in. This can be a word list, word sequences, and word permutations.

vocabulary only GO words: this file contains the results of the searching process.

file of GO words: this file contains the search patterns that are searched for. If the search patterns are not in a file, one can enter them interactively.

Invoking the program out of the supervisor:

```
wobana kontakt.wb -g
```

6.6.2 Information messages

```
INTEXT/586 version 4.0 - 2/1997 - 8.02.1997 11:50
```

```
  routine: WOBANA
```

```
  application: searching vocabularies
```

```
input file      kontakt.wb
```

```
output file     kontakt.wbg
```

```
statistics:
```

type of string	types	token	TTR
strings read:			
words	259	647	0.400
digits	0	0	0.000
other	3	94	0.032
total	262	741	0.354

```
strings written:
```

words	4	6	0.667
digits	0	0	0.000
other	0	0	0.000
total	4	6	0.667

```
differences (read - written strings):
```

type of string	types	%	token	%
words	255	98.456	641	99.073
digits	0	0.000	0	0.000
other	3	100.000	94	100.000
total	258	98.473	735	99.190

```
WOBANA start: 11:50:30
```

```
WOBANA end: 11:51:07
```

```
WOBANA needed 37 seconds CPU-time
```

6.7 TTR dynamics

TTR dynamics are calculated only for strings that have a letter or a digit as first character. After each token the value of the TTR is recalculated. Sampling is supported. The output file consists of the token, the cumulated values for types, tokens and the TTR after each token. TTR dynamics show the growth of the vocabulary of a text and supplement the pure TTR which is dependent on text size. The value of the TTR starts with 1 and decreases in general, sometimes it increases. Useful for interpretation are the increases and the number of tokens if certain values are reached, especially for the comparison of texts.

6.7.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 15.05.1997 09:55	
program: WORDBOOK	
application: TTR-dynamics	
14935 RAM free	input/output files
name of system file	kontakt.itx
file of TTR-dynamics:	kontakt.ttr
parameters	
process all text units	yes
case folding enabled	yes
inclusion of types	no
number of decimal digits	3
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

file of TTR-dynamics: the name of the file where the TTR-dynamics are written to. The name may contain drive and/or directory specifications.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed (see chapter 5 on page 59).

case folding enabled Letters can be treated as the same, if they are different only in their case (lower or upper case).

inclusion of types: If denied, the types are not written to the output file, which reduces its size dramatically. Also the data can be processed more easily with Gabriel Altmann's programs, e.g. Altmann Fitter to analyse the distribution.

number of decimal digits: the precision of the TTR-values can be specified, default is 3 digits.

Invoking the program out of the supervisor: `wb kontakt.itx kontakt.ttr -TTR`

6.7.2 Information messages

INTEXT/586 Version 4.0 - 4/1997 - 26.04.1997 20:30

routine: WORDBOOK

application: TTR dynamics

input file kontakt.itx

output file kontakt.ttr

case folding enabled

sort table ISM.DEF used

statistics:

strings (token) read:

- I 01: 560 text units
- I 02: 6157 words
- I 03: 414 numbers
- I 04: 2258 other
- I 05: 8829 total of strings
- I 06: 10.995 words/line
- I 07: 0.739 numbers/line
- I 08: 4.032 other/line
- I 09: 15.766 total of strings/line
- I 25: 420 KB RAM used
WORDBOOK started: 20:30:10
WORDBOOK ended : 20:30:23
WORDBOOK needed 13 seconds CPU-time

6.7.3 Results of TTR dynamics

type	token	TTR	strings
1	1	1.00	Junger
2	2	1.00	Mann
3	3	1.00	mit
4	4	1.00	gutem
5	5	1.00	handwerklichen
6	6	1.00	Beruf
7	7	1.00	möchte
7	8	0.88	mit
8	9	0.89	einer
9	10	0.90	Frau
10	11	0.91	zwischen
11	12	0.92	30-40j
13	14	0.93	gerne
13	15	0.87	mit
14	16	0.88	Kindern
15	17	0.88	aus
16	18	0.89	dem
17	19	0.89	Raum
18	20	0.90	Köln-Leverkusen-Gladbeck
19	21	0.90	eine
20	22	0.91	schöne
21	23	0.91	Freundschaft
22	24	0.92	aufbauen
23	25	0.92	Wenn
24	26	0.92	das
25	27	0.93	auch
26	28	0.93	ihr
27	29	0.93	Wunsch
28	30	0.93	ist
29	32	0.91	dann
30	33	0.91	schreiben
31	34	0.91	sie
32	35	0.91	mir
33	36	0.92	doch
34	37	0.92	mal
36	39	0.92	Jammerschade
37	41	0.90	Stell
38	42	0.90	dir

6.8 The use of search patterns

Search patterns define a category system. They are organised in a file of search patterns together with codes and parameters. In a content analysis the search patterns are searched for within every text unit. If it is found, the code that belongs to the search pattern is written to the output file(s). The parameters are to be specified in the parameter field and control the features for the validity of the coding (files for uncoded, coded, and negated text units).

There are two kind of search patterns:

1. string or any word of it, also parts of words and word sequences
2. word root chains

Search patterns can be words or parts of it, but also letters or syllables. Every search pattern starts with a colon (':') in column 7 and ends with a colon. Both colons must exist. Instead of a colon any other character that does not occur in the search pattern may be used (separator). The columns 1-3 can be used for generating a concordance and must be used for a three digit code for a content analysis. Columns 4-6 are called the parameter field where parameters can be specified. These control the output of rapport files for ambiguous, uncoded and/or negated text units.

In all INTEXT-versions the number of search patterns that can be processed in one analysis is limited by the available memory (RAM), only the number of word root chains is limited to 1000.

6.8.1 Specifications in the parameter field

The parameter field can be used to control the treatment of each search pattern. The following parameters are possible:

C coding control All text units that contain the search patterns are written to the file of coded text units. If interactive coding is enabled, the text unit, the search pattern, the category number and the corresponding label are displayed. You are asked whether and if yes, which code is to be used.

U Uppercase All characters of the search pattern are translated into uppercase, so that lower case and upper case are treated as the same. This is useful with words that are capitalised because they are at the beginning of a sentence.

N negation The search pattern is checked for negation. If an odd number of indicators before and after the search pattern occurs (default: 2), the search pattern is not coded. The search pattern is coded when an even number (e.g. double negation – litotes) of indicators occurs.

6.8.2 Strings

Strings as search patterns are a part of a text unit. It doesn't matter whether a string is just a letter or a sequence of words. Strings may also be any part of a word. The maximum length is 200 characters. Within a string the ? can be used as a wildcard character, the use is the same as in MS-DOS file names. A ? substitutes exactly one character. The asterisk * is the wildcard character for any number of characters before and after it, but is limited to one single word

A line in the file of search patterns (DIC-file) is structured as follows:

column	contents
1 - 3	code
4 - 6	parameter field (may be left blank)
7	separator (e.g. inverted comma)
8 - 200	search pattern (delimited with separator)

An example for the definition of strings as a search pattern (with option U enabled for ignoring differences in case):

search pattern	found text
' Politik '	Politik (no other words)
' Politik'	Politik, Politiker, Politikwissenschaft
'politik '	Politik, Finanzpolitik, Ostpolitik
'politik'	Politik, Justizpolitiker, Metropolitikone
' ab'	abschreiben, ablesen, abkaufen
' H?us'	Haus, Häuser, Heuss
' Ver????ung '	Versuchung, Verbleiung, Verletzung
' Ver*ung '	Versuchung, Verbleiung, Verletzung, Verbeamtung

Example for a string as a search pattern:

```
001 C 'umwelt'
002 C ' Umwelt'
013 C 'vergiftung '
```

By using blanks one can define whether a string should be treated as a word or as a part of it. So it's possible to define unambiguous words or parts of words as prefixes or suffixes. The examples mentioned above show the use, more examples are in the provided *.DIC files (e.g. KONTAKT.DIC).

6.8.3 Word root chains

Word root chains are similar to strings as search patterns. Word roots – that are parts of strings – can be defined that must occur within a text unit. in the order they are specified.

The distance between two word roots doesn't matter. The distance between the word root as well as the order within a text unit can be varied. There are three kinds of word root chains that must be marked in the parameter field:

- option D: direct mode. The word roots must occur in words that follow each other without any other strings (words, colon etc.) between them within a text unit.
- option F: following mode. The word roots must follow each other within a text unit, but the distance between them doesn't matter and is dependent on the definition of a text unit.
- option S: simultaneous mode. The word roots must occur within a text unit, order and distance do not matter.

The definition of word root chains is done with the (<,>) symbols. Before and after the word root may be characters, but there are non required. < indicates, that characters in front of the word root are allowed, > indicates, that characters after the word root are allowed. Also the wild card symbols ? and * may be used, the same rules as for strings as search patterns apply. Up to 1000 word root chains can be used in one analysis.

Examples:

word root chain	found text
'<schlecht> Politik'	schlechte Politik, schlechter Politik schlechte und dumme Politik
'<schlecht> Politik>'	schlechte Politiker, schlechter Politiker schlechte, dumme und teure Politiker
'<schlecht> <politik>'	sauschlechte Wirtschaftspolitikziele
'<gift> Umwelt '	giftige Umwelt, vergiftete und dreckige Umwelt

Examples for a word root chain as a search pattern:

```
004 C '<gift> Umwelt'
```

```
005 C 'sauber> <umwelt>'
```

```
005 C 'sauber> Umwelt>'
```

6.8.4 Printed output of a category system

```
c ' uncontrollable'
cu' tense'
f 'bully shot>'
u ' surprise'
u ' try'
```

Category 11: age

```
' older'
' child
'age'
```

Category 12: Sex, love

```
' marriage'
' boy friend'
' girl wonder'
' sex appeal'
' girl friend'
' petting'
' tender' also code 9
c ' sex'
c ' darling' also code 9
c ' masculin'
c 'love'
```

Category 13: science, progress

```
' computer'
c ' PC'
' CeBit'
' Concorde '
```

6.9 Concordances

Two formats are available for concordances, both write to an output file that can be processed by other INTEXT programs or by third party programs. Words, word sequences and word root chains can be search patterns, details are described in chapter 6.8 on page 94.

- short concordance: every line contains a search pattern, 79 characters is the default line length. The search pattern can be emphasised by a (print) attribute. An alphabetical sorting of the file is not required.
- long concordance: every search pattern is written separately, they are separated by blank lines, between those there are the concordances. The search pattern can be emphasised by a (print) attribute. An alphabetical sorting of the file is required.

With IRM these concordances can be output in short format – looking like a KWIC (KeyWord-In-Context) – or in long format – similar to a KWOC (KeyWord-Out-of-Context). A concordance has a variable length. Using short concordance format printing is linewise. Using long concordance format every search pattern is printed on a separated line, and if IRM is used for viewing or printing, the search patterns can be emphasised (**bold**, *in italics* or underline).

6.9.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 15.05.1997 09:27	
program: SUWACO	
application: short concordance	
input/output files	
name of system file	kontakt.itx
file of search patterns	kontakt.dic
file name of short concordances	kontakt.sis
parameters	
process all text units	yes
line length	79
interactive selection	no
inclusion of external variables	no
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

file of search patterns: the name of the file where the search patterns are stored (DIC-file). The number of search patterns is discussed in chapter 6.8 on page 94. The name may contain drive and/or directory specifications.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed (see chapter 5 on page 59).

file name of short concordances: the file name the concordances are written to.

line length: The default value for short concordances is 79 characters, for long concordances it is 131. The value is dependent on the output medium (screen or printer). The context can be enlarged if external variables are not included.

interactive selection: yes means, that each occurrence requires an answer whether it is to be written to the output file or not. No means, that all occurrences are included.

inclusion of external variables If this question is denied, the concordances are only written together with their codes, the external variables are suppressed.

After the generation of the concordances the output file is sorted by alphabet, the MS-DOS sort is used for that. Only files smaller than 64 KB can be sorted, other sorts can be used if they are configured (menu External programs).

Start the program from MS-DOS level: `suwaco kontakt.itx kontakt.dic`

6.9.2 Information messages

```
INTEXT/586 Version 4.0 - 4/1997 - 15.05.1997 16:02
  routine: SUWACO
  application: short concordance
input file      kontakt.itx
category file   kontakt.dic
concordance file kontakt.SIS
- C 01:        1363 search units processed
- C 05:         194 with option C marked search patterns
- C 06:       1148 with option U marked search patterns
- I 01:         560 text unit read
- I 02:       8858 words read
- I 15:       1997 output records in SIC file
SUWACO start:  16:02:22
SUWACO end:    16:08:18
SUWACO needed 356 seconds CPU time
```

6.9.3 Printed output of a concordance (short form)

28 333 2 1 nst (44-164) Witwer , mit **10**jährigem Sohn möchte , *
 28 422 1 1 nsgröße 36 und Mutter eines **12**jährigen Sohnes , alles
 28 324 1 1 n , Lehrerin und habe einen **5**jährigen Sohn ; ich mag :
 28 301 1 1 liche Lehrerin (33 , 174 , **7**jährige Tochter) sucht f
 28 413 1 1 gute Figur , 35 Jahre , mit **9**jähriger Tochter , sucht
 3 326 2 3 d anderswo Lieber hätte ic* **Aachen-Köln-Düsseldorf-** un
 26 227 1 1 wrne nach einem gemütlichen **Abend** zu zweit , im Sommer
 26 119 2 1 g Stille , Natur und ruhige **Abende** genauso wie Theater
 19 148 2 1 e ! 20 Jahre und noch keine **Abnutzungserscheinungen** !
 15 109 2 1 * **Adam** sucht *
 44 426 2 1 de (25) , Student , weder **Adonis** noch Quasimodo , gr
 41 326 2 3 Studium oder einer anderen **Aktivität** kennengelernt ,
 16 405 1 1 aktive , studierte Frau su* **Alleinsein** ist doof ! Attr
 16 438 2 3 is zweimal in der Woche des **Alleinseins** müde ist . Dis
 1 217 2 1 htraucher , 172 , mit guter **Allgemeinbildung** sucht *ch
 27 411 1 2 t Schnäuzer) ab und zu den **Alltag** vergessen . *rn mit
 27 301 1 2 ie kleinen Widrigkeiten des **Alltags** lächeln kann . *di
 32 215 2 2 liche Frau , entsprechenden **Alters** , auch mit Kind , k
 32 245 1 2 tten Partner entsprechenden **Alters** . * * ne
 32 442 2 2 eutsche Partnerin passenden **Alters** . Bitte keine Femin
 32 339 2 2 ge Ehepartnerin , passenden **Alters** für Neuanfang . *ig
 44 104 2 1 * Chiffre : **Andre Heller** *
 28 323 1 1 arkett gewachsen ist , ohne **Anhang** , blond , und sehr
 28 208 2 1 ch-häßliche Jahre alt , mit **Anhang** , finanziell unabhä
 28 442 2 2 eministin und Emanze , ohne **Anhang** , gern auch jünger
 28 335 2 1 gesund , vorzeigbar , ohne **Anhang** , sehr reisefreudig
 26 418 1 2 nen , lebensfrohen Mann zum **Anlehnen** , der einfühlsam
 26 305 1 2 ärtlichkeit , Verständnis , **Anlehnung** , Gedankenaustau
 19 201 2 1 Jahre , römischkatholisch , **Apathiker** , rechtseitig te
 39 330 2 1 , liebt Reisen , gepflegte **Atmosphäre** , das Meer und
 18 405 1 1 u su* **Alleinsein** ist doof ! **Attraktive** , studierte Fra
 18 233 1 1 r , 27 Jahre , möchte Mann* **Attraktive** Krankenschweste
 18 321 1 1 , studiert , schlank , spo* **Attraktive** Sie , 36 , 177
 18 320 1 1 /167 , kultiviert und femi* **Attraktive** Skorpionin , 49
 18 232 1 1 65 , schlank , blond , suc* **Attraktive** Vierzigerin , 1
 18 301 1 1 ehrerin (33 , 174 , 7jäh* **Attraktive** und fröhliche L
 18 145 2 1 er Er , 28 Jahre , 184 Zen* **Attraktiver** , selbstbewußt
 18 124 2 1 ine Lust auf konventionell* **Attraktiver** Student hat ke
 18 211 2 1 nk mit allen viel zitierten **Attributen** ausgestattet ,
 27 244 1 3 Partnerschaft . * * zwecks **Aufbau** einer harmonischen
 30 332 2 1 it meinem Kind im südlichen **Ausland** (gute Verbindung
 30 328 2 1 iegenden Teil des Lebens im **Ausland** verbringt . Zur Ze
 30 328 2 2 e Lebensgefährtin , die das **Auslandsleben** liebt , bei
 31 108 2 2 t ganz so jung , gerne auch **Ausländer**) denen es zur Z
 31 214 2 2 lig , kein Hindernis , auch **Ausländerin** angenehm . *11
 18 414 1 2 ein interessanter Mann mit **Ausstrahlung** , Geist und N

6.10 Search patterns in the text unit

Search patterns in text unit are similar to concordances, the context however is not limited by a number of characters but is the whole text unit. All kinds of search patterns are possible, details are described in the chapter on the definition on search patterns on page 94. The results are written to the output file and can be processed by other programs.

The output file consists of lines that start with the search pattern. After a blank the whole text unit follows. Displaying the results with IRM programs the search patterns can be underlined, **bold face** or in *italics*.

6.10.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 23.05.1997 09:11	
program: SUWACO	
application: search units in text unit	
input/output files	
name of system file	kontakt.itx
file of search patterns	kontakt.dic
file name of search patterns in text unit	kontakt.sit
parameters	
process all text units	yes
interactive selection	no
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

file of search patterns: the name of the file where the search patterns are stored (DIC-file). The number of search patterns is discussed in chapter 6.8 on page 94. The name may contain drive and/or directory specifications.

file name of search patterns in text unit: the name of the output file that contains the search patterns in text unit.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed (see chapter 5 on page 59).

interactive selection: yes means, that each occurrence requires an answer whether it is to be written to the output file or not. No means, that all occurrences are included.

Afterwards the output file is sorted by alphabet, the sort program specified in the configuration is used, default is the sort program of MS-DOS. This only works with files smaller than 64 KB in size. One can use other sort programs if they use the same syntax to specify sort keys, e.g. QSORT by Ben Baker.

Start the program from MS-DOS level: `suwaco kontakt.itx kontakt.dic -c -t -sit`

6.10.2 Information messages

```
INTEXT/586 Version 4.0 - 4/1997 - 23.05.1997 16:02
  routine: SUWACO
  application: search patterns in text unit
input file      kontakt.itx
category file   kontakt.dic
SIT-file        kontakt.sit
- C 01:         1363 search units processed
- C 05:         194 with option C marked search patterns
- C 06:         1148 with option U marked search patterns
- I 01:         560 text unit read
- I 02:         8858 words read
- I 15:         1997 output records in SIT file
SUWACO start:   16:02:22
SUWACO end:     16:05:44
SUWACO needed 202 seconds CPU time
```

6.11 Content analysis

A content analysis processes search patterns. These can be words, part of words, word sequences or word root chains. The results are written to files and can be processed by other programs. The same rules for search patterns and search patterns described in the last chapter apply.

6.11.1 Category labels

The category labels support the documentation of codes and their meaning. Definitions of category labels are compulsory, the definitions must be stored in a file. Each line of this file contains – starting on column 1 – the number of the category. A blank follows, and after the blank you write the category label for the category. The maximum length is 60 characters, if it is longer, it is truncated. Up to 999 category labels are possible.

Example for a file with category labels:

```
1 character
2 inner values
3 attractiveness
4 intellectual mobility
```

The file KONTAKT.LAB contains another example.

6.11.2 Category system

A content analysis is based on a category system that consist of search patterns stored in a file (DIC-file). It can be generated as follows:

- Generating a category system with an editor Within the supervisor or also from DOS an editor can be used to generate a category system. The editor – or a text processor – must only be able to store a file unformatted (pure ASCII file). Please note that the rules for the search patterns must be followed, especially the positions of the code, the parameter field and the search pattern. More details can be found in the chapter on search patterns on page 94.
- Interactive generation of a category system

For every string in a vocabulary it can be decided, if and with which code this string is to be coded. Also the specifications for the parameter field – enabling/disabling case folding and potential ambiguity – can be entered. A DIC-file for the search patterns and a LAB-file for the category labels are generated.

INTEXT/586 4.0 - 4/1997 - 18.04.1997 09:04	
routine: WOBANA	
application: generate category system	
vocabulary sport.wb	category system sport.lab
freq string	code category labels
3 started	1 negative action
Write to output file? (y/n)	2 negative feeling
	3 part of body
	4 kind of sport
	5 negotiations
	6 dangers
options	
case folding enabled	no
ambiguous	no
line 213	

The screen is divided into two parts: on the right you see the already existing codes with their category labels, in the upper left corner you see the frequency and the string you have to decide whether it belongs to the category system or not. The following commands are possible:

- n: no, show next string from vocabulary
- y: yes, the code for the string must be entered, new codes require the input of a category label. Options for case folding and ambiguity (option U and C in the parameter field) can be specified. Search pattern and category label are written to the output files. The string cannot be edited.
- s: stop, stop and restart with the last processed string. The file for search patterns and category labels are written when stopped. After a restart both files are read, and the category labels are shown on the screen. New search patterns and category labels are appended to their files.

Both files can be edited, this makes sense because only whole strings from the vocabulary, but no parts of them, can be selected.

6.11.3 Results of the coding

The content analysis is based upon the fact that search patterns are looked for in each text unit. This is called coding. If a search pattern is found, its code will be processed further on. The possibilities to define search patterns are described in chapter 6.8 on page 94. The results are written into the appropriate output files and can be analysed with statistical software; a setup for SAS, SPSS, and ConClus can be generated.

The coding results can be written to the output file in two modes:

- vector file: the codes are written to the output file in the order they occur within the text unit.
- tabulation file: for each code there is a counter that holds the frequency for the code in the text unit. These counters are written to the output file after each text unit. The size of the tabulation file is calculated from the number of categories of the category system, each counter must not exceed 999 within a text unit.

The codes of both files may have up to three digits (values 1 to 999). If this limit is exceeded, an error message is displayed providing more information on the text unit. The coding does not take the context into account, so that ambiguities of search patterns and negations are not recognised and can result in erroneous codings. Therefore it is possible that potential ambiguous and/or negated search patterns can be coded interactively.

The validity of the coding process can be controlled by interactive coding or by rapport files:

- file of the coded text units: all text units containing at least one search pattern of the category system is written to this output file. Category labels can be written behind each coded part of the text, this is useful for the validation of the coding process.
- file of uncoded text units: all text units that do not contain a search pattern of the category system are written to this output file.
- file of negated text units: all text units containing at least one search pattern of the category system where negation indicators before or after the search pattern occurred in the specified distances are written to this output file.
- file of coded search patterns: each coded search pattern is written to the output file with external variables, code, text, and category label. This file can become very large.
- file of overlapping text segments: text segments where at least one character is part of several search patterns. This causes problems with the vector file, not all codes can be displayed. The reasons may be technical or caused by the category system.

6.11.4 Interactive coding

The following figure shows the screen while coding interactive: the current text unit with the external variables, an already coded search pattern (tender) with its category (26) and its label. The search pattern to be coded is displayed in red (here printed bold). At the bottom of the screen search pattern, code and category label are displayed, together with the available commands. The gap between top and bottom of the screen is filled with the category system.

Interactive coding with category system:

```

text unit: 160188 Observer man woman partner
warm-hearted woman . Are you a catholic Christian , look for tender[26: partnership
erotic behaviour] ness and cosyngness (marriage)
20 blue collar professions          21 bureau professions
22 social professions                25 positive inner values
26 partnerschip - erotic behaviour  27 partnership - neutral
28 family orientation                30 nature, holidays, travelling
31 nationality                        32 age 17-30 years
33 age 31-45 years                   34 age 46-60 years
35 age 60+ years                     36 practical abilities
37 enjoyment (not sexual)            38 health orientation
39 prestige

search pattern: ' MARRIAGE'
category 28: family orientation
code? (n=no ■ y=yes■ S=no + stop ■ s=yes + stop
same code: press RETURN, change code: enter code

```

The first decision is, whether to code or not, with **F1** you can display the category system with its labels, browsing through it is done with **↓** and **↑**. The following commands are available:

- b – show the same search pattern again, coding can be rejected.
- j – search pattern is coded, code can be accepted or changed.
- n – search pattern is not coded.
- s – search pattern is coded, the code can be accepted or changed. After the last search pattern the results are written to the output files, and the coding will be terminated.
- S – search pattern is not coded, code can be accepted or changed. After the last search pattern the results are written to the output files, and the coding will be terminated.

Interactive coding can last a long time. Therefore it is possible to discontinue the coding and continue later. After the appropriate command was issued, the remaining search patterns are coded and the results written to the output files. After a restart the coding is continued where it was stopped, the results are appended to the appropriate files. Another termination is possible.

The coding suggestion does not consider negation. Also an extension of the category system with new codes is **not** possible.

Considering uncoded and not as suggested as originally intended from the category system coded search patterns a coefficient (Interactive coding reliability coefficient) is computed:

$$ICRC = \frac{\text{coded search patterns} - \text{rejected search patterns} - \text{changed search patterns}}{\text{coded search patterns} + \text{rejected search patterns}}$$

The range is between 0 and 1. The higher it is, the better the reliability is.

Control of the coding process: results

ID 1	ID 2	ID 3	Code	search entry	found text	category label
160188	400112	1	18	J?NGE	Junge	[körperliche Merkmale]
160188	400112	1	36	handwerk	handwerk	[handwerkliche Fähigkeiten]
160188	400112	2	28	KIND	Kind	[familiäre Orientierung]
160188	400112	2	33	40J	40j	[Alter 31-45 Jahre]
160188	400112	4	3	Köln	Köln	[örtliche Gebundenheit]
160188	400112	4	3	RAUM	Raum	[örtliche Gebundenheit]
160188	400112	3	27	FREUNDSCHAFT	Freundschaft	[partnerschaftliches Verhalten - neutral]
160188	400112	3	27	AUFBAU	aufbau	[partnerschaftliches Verhalten - neutral]
160188	400212	1	32	29J	29j	[Alter 17-30 Jahre]
160188	400312	2	18	J?NGE	junge	[körperliche Merkmale]
160188	400312	2	18	MÄD	Mäd	[körperliche Merkmale]
160188	400312	2	32	23J	23j	[Alter 17-30 Jahre]
160188	400312	3	27	GESPRÄCH	Gespräch	[partnerschaftliches Verhalten - neutral]
160188	400312	3	30	AUSFLÜGE	Ausflüge	[Natur, Urlaub, Reisen]
160188	400312	1	7	AKADEMI	Akademi	[akademische Berufe]
160188	400312	1	27	EINFÜHL	einühl	[partnerschaftliches Verhalten - neutral]
160188	400312	1	27	RÜCKSICHT	rücksicht	[partnerschaftliches Verhalten - neutral]
160188	400312	4	18	Bildzu	Bildzu	[körperliche Merkmale]
160188	400402	1	4	GEIST	geist	[geistige Mobilität]
160188	400402	1	12	VITAL	vital	[körperliche Fitness]
160188	400402	1	18	BLOND	blond	[körperliche Merkmale]
160188	400402	1	18	SCHLANK	schlank	[körperliche Merkmale]
160188	400402	1	35	65J	65j	[Alter 60+ Jahre]

Optional a setup for statistical analyses for SAS/PC, SPSS/PC+ and ConClus is generated. It consists of the data definition part, the labels for the variables, and the commands for calculating frequencies.

6.11.5 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 15.05.1997 09:32			
program: SUWACO			
application: content anaylsis			
name of system file			kontakt.itx
process all text units			yes
file of search patterns			kontakt.dic
file of category labels			kontakt.lab
file of codes as counters			kontakt.tab
file of codes in their sequence			no
setup for SPSS			kontakt.sps
Specification of the interactive coding			
type of search patterns	ICod protocol files		options
all search patterns	no no		unique
ambiguous search patterns	no no		
negated search patterns	no no		2 2
overlapping search patterns	no no		overwrite
uncoded text units	no		
F1 help files	F2 help parameters	F3 change	F4 start F10 end

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed (see chapter 5 on page 59).

file of search patterns: the name of the file where the search patterns are stored (DIC-file). The number of search patterns is discussed in chapter 6.8 on page 94. The name may contain drive and/or directory specifications.

file of category labels: this file contains the category system with codes and their labels. The name may contain drive and/or directory specifications.

file of codes as counters: the name of the file where the counters of the categories are stored. The name may contain drive and/or directory specifications.

file of codes in their sequence: the name of the file where the codes in their sequence are stored. The name may contain drive and/or directory specifications.

number of codes within a text unit: If a file of codes in their sequences is requested, this number specifies how many patterns are coded within a text unit. The value is dependent on the longest text unit. If this number is exceeded, a warning is given.

Coding continues without storing in this file. The statistics concerning the found search patterns are correct although the VEC-file is incorrect.

setup for: A complete setup for further processing of the generated raw data matrix with statistical packages is generated for SPSS/PC+, SAS/PC or ConClus/PC. It contains the reading specifications (data list), the labels for the codes (var labels) and the commands for frequency tables. The file is named *.SPS for SPSS, *.SAS for SAS, and *.STK for ConClus.

coding parameters For each type of search patterns coding parameters can be specified:

- ICode: yes: these search patterns are coded interactive; no: automatic coding.
- protocol files: protocol files for these search patterns are requested. If a file is not to be omitted, delete the file name.
- options: here parameters for the types of search patterns can be defined.
 - all search patterns: unique or ambiguous.
 - * unique: the ambiguity of all search patterns is treated as specified in the parameter field. If a protocol file for all search patterns was requested, only the text units containing at least one potential ambiguous search pattern are written to the output file. If interactive coding is enabled, only the marked search patterns are coded interactive, all others are coded automatic.
 - * ambiguous: if interactive coding is enabled, all search patterns are coded interactive, useful for teaching purposes or pretests. If a protocol file was requested, all text units that contains at least one search pattern are written to this file.
 - ambiguous search patterns; with or without labels.
 - * with labels: labels are useful for the coding control of all or of the ambiguous search patterns, after them code and category label follow.
 - * without labels: if the file is used for further processing (e.g. generate a word list) category labels disturb.
 - negated search patterns: distance of negation. Two values can be specified: the first one specifies the number of strings before the search pattern is searched for negation indicators, the second one specifies the number of strings after the search pattern is searched for negation indicators. The negation indicators are counted. If the number is odd, a negation exits, even numbers indicate a double negation (litotes). 0 means to disable negation control for all search patterns.

Start the program from MS-DOS level: `suwaco kontakt.itx kontakt.dic`

6.11.6 Information messages

```

INTEXT/586 Version 4.0 - 4/1997 - 26.05.1997 16:09
      routine: SUWACO
      application: content analysis
input file      kontakt.itx
category file   kontakt.dic
tabular file    kontakt.TAB
CODED-file     kontakt.ctx
REST-file      kontakt.rtx
file of negations kontakt.ntx
label file      kontakt.lab
job for SPSS/PC+ in file kontakt.sps
- C 01:        1363 search units processed
- C 05:         194 with option C marked search patterns
- C 06:        1148 with option U marked search patterns
- I 01:         560 text units read
- I 02:        8858 words read
- I 11:        1996 coded search units in TAB file
- I 13:         83 coded text unit in CODED file
- I 14:         62 uncoded text unit in REST file
- I 22:        560 output records in TAB file
- I 23:        560 coding units
- I 24:         62 uncoded coding units
- I 25:        498 coded coding units
- I 26:         19 overlapping search patterns
SUWACO start:  16:09:02
SUWACO end:    16:15:01
SUWACO needed 359 seconds CPU time

```

6.12 List of uncoded words

In the phase when you develop a category system it is useful, which strings of the word list are not found by search patterns and therefore remain uncoded. These are contained in the word list of uncoded words, which is only useful if the search patterns are single words.

6.12.1 Parameters of the program

INTEXT/586 4.0 - 2/1997 - 06.02.1997 15:03					
program: WOBANA					
application: uncoded words					
input/output files					
file of word list					kontakt.wb
file name of uncoded words					kontakt.rwb
file of search patterns					kontakt.dic
F1	help files	F2	help parameters	F3	change
				F4	start
					F10 end

file of search patterns: the name of the file where the search patterns are stored (DIC-file). The number of search patterns is discussed in chapter 6.8 on page 94. The name may contain drive and/or directory specifications.

file of category labels: this file contains the category system with codes and their labels. The name may contain drive and/or directory specifications.

file name of uncoded words: this file contains the string that will not be coded during a content analysis. It can be used to improve the category system.

There are no parameters.

Start the program from MS-DOS level:

```
wobana kontakt.wb kontakt.dic -r
```

6.12.2 Information messages

INTEXT/586 4.0 - 4/1997 - 28.05.1997 09:55

program: WOBANA

application: list of uncoded words

input file kontakt.wb
output file kontakt.rwb
GO file kontakt.dic

statistics:

type of string	types	token	TTR
strings read			
words	6095	41098	0.148
numbers	588	3806	0.154
other	35	17605	0.002
total	6718	62509	0.107

strings written:

words	3359	27291	0.123
numbers	278	1767	0.157
other	30	17600	0.002
total	3667	46658	0.079

differences (read - written strings):

type of string	types	%	token	%
words	2736	44.889	13807	33.595
numbers	310	52.721	2039	53.573
other	5	14.286	5	0.028
total	3051	45.415	15851	25.358

WOBANA start: 09:55:17

WOBANA end: 10:00:28

WOBANA needed 311 seconds CPU time

6.13 Test on multiple search patterns

This test checks, whether a search pattern is a part of another or if it occurs more than once. If this is the case, the danger of multiple coding arises which leads to weighting and biasing the results. This time consuming test is done with the category system and also tests, whether parts of word roots occur in other search patterns. It lasts quite long, the results can be routed to the screen and/or to a file.

6.13.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 15.05.1997 09:52	
program: SUWACO	
application: multiple search patterns test	
input/output files	
file of search patterns	kontakt.itx
file name for category labels	kontakt.lab
file name of multiple search patterns	kontakt.dse
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

file of search patterns: the name of the file where the search patterns are stored (DIC-file). The number of search patterns is discussed in chapter 6.8 on page 94. The name may contain drive and/or directory specifications.

file of category labels: this file contains the category system with codes and their labels. The name may contain drive and/or directory specifications.

file name of multiple search patterns: this file contains all search patterns with their codes and labels, that occur more than once within the category system (DIC-file) or are part of another search pattern.

There are no parameters.

Start the program from MS-DOS level:

```
SUWACO kontakt.dic kontakt.lab -pd
```

6.13.2 Results of the multiple entry test

line		search pattern	code	category label
519	<	WITW>	16	[Lebenseinschnitte]
311	<	BEAMTENWITW>	8	[hoher ökonomischer Status]
445	<	UFO>	15	[Metaphorik]
395	<	UFO>	13	[Metaphysik]
614	<	BIENE>	17	[Sex]
411	<	BIENE>	15	[Metaphorik]
537	<	BRIEFMARK>	17	[Sex]
413	<	BRIEFMARK>	15	[Metaphorik]
861	<	KATER >	26	[part. Verhalten - erotisch getönt]
425	<	KATER >	15	[Metaphorik]
862	<	KATZE >	26	[part. Verhalten - erotisch getönt]
426	<	KATZE >	15	[Metaphorik]
543	<	KOKOSNÜSSEKNACKEN >	17	[Sex]
427	<	KOKOSNÜSSEKNACKEN >	15	[Metaphorik]
612	<	MISSIONAR>	17	[Sex]
429	<	MISSIONAR>	15	[Metaphorik]
1294	<	NACHTEULE>	41	[gesellschaftliche Aktivität]
432	<	NACHTEULE>	15	[Metaphorik]
151	<	RHEIN>	3	[örtliche Gebundenheit]
440	<	RHEINLÄNDER>	15	[Metaphorik]
1031	<	UNTER DIE HAUBE >	28	[familiäre Orientierung]
447	<	UNTER DIE HAUBE >	15	[Metaphorik]

6.14 Readability analysis

REFO computes 8 different formulas that are based on syntactic criteria. Implications of the most formulas are that they are language and/or text genre specific, so the results have to be interpreted carefully. In opposite to the literature mentioned in the footnote, REFO doesn't work with a sample of 100 words, but with the whole text or parts of it (see chapter 5 on page 59). The values are between 0 and 100 (REI 120). The higher the value is, the better the readability is. If the values are out of range, it is very likely that the formula is used on texts it has not been developed for.

→The text unit must be the sentence.

The formulas are calculated as follows:

$$REI = 206.835 - \left(\frac{\text{number of syllables}}{\text{number of words}} * 0.864\right) - \left(\frac{\text{number of words}}{\text{number of sentences}}\right) \text{ (Flesch 1948)}$$

$$MREI = -2.2029 + \left(0.445 * \left(\frac{\text{number of syllables}}{\text{number of words}}\right)\right) + \left(0.778 * \left(\frac{\text{number of words}}{\text{number of sentences}}\right)\right) \text{ (Powers 1958)}$$

$$NREI = (1.5999 * \text{monosyllable words}) - \left(1.015 * \left(\frac{\text{number of words}}{\text{number of sentences}}\right)\right) - 31.517 \text{ (Farr, Jenkins, Paterson 1951)}$$

$$MNREI = 8.4335 + (0.0648 * \text{monosyllables words}) + \left(0.923 * \left(\frac{\text{number of words}}{\text{number of sentences}}\right)\right) \text{ (Powers 1958)}$$

$$TRI = (0.449 * \text{monosyllable words}) - (2.467 * \text{punctuation marks}) - (0.937 * \text{foreign words}) - 14.417 \text{ (Kuntzsch 1981)}$$

$$DB1 = \left(1.0364 * \left(\frac{\text{number of words} - \text{number of punctuation marks}}{\text{number of words} - \text{number of sentences}}\right)\right) + \left(0.0194 * \left(\frac{\text{number of words}}{\text{number of sentences}}\right)\right) - 0.6059 \text{ (Danielson/Bryan 1963)}$$

$$DB2 = 131.059 - \left(10.364 * \left(\frac{\text{number of words} - \text{number of punctuation marks}}{\text{number of words} - \text{number of sentences}}\right)\right) + \left(0.194 * \left(\frac{\text{number of words}}{\text{number of sentences}}\right)\right) \text{ (Danielson/Bryan 1963)}$$

$$AVI = 180 - \left(\frac{\text{number of words}}{\text{number of sentences}}\right) + \left(\frac{\text{number of syllables}}{\text{number of words}}\right) * 58,5 \text{ (Amdahl 1978)}^6$$

For the TRI index it is possible to specify strings as indicators for foreign words. The

⁶Literature: Amstad, T. (1978): Wie verständlich sind unsere Zeitungen? Dissertation, Zürich. Ballstaedt, Steffen-Peter; Heinz Mandel; Wolfgang Schnotz; Sigmar-Olaf Tergan (1981): Texte verstehen, Texte gestalten. München, p. 212. Danielson, Wayne A; Sam Dunn Bryan (1963): Computer Automation of Two Readability Formulas. In: Journalism Quarterly 40, p. 201-206. Flesch, Rudolph (1948): A New Readability Yardstick. In: Journal of Applied Psychology 32, p. 221-233. Farr, J.N.; J.J. Jenkins; D.G. Paterson (1951): Simplification of Flesch Reading Ease Formula. In: Journal of Applied Psychology 35, p. 333-337. Kuntzsch, Michael (1978): TRI – Wie man die Verständlichkeit von Nachrichten erhöhen kann. M.A. thesis, Mainz. Merten, Klaus (1983): Inhaltsanalyse. Einführung in Theorie, Methode und Praxis. Opladen, p. 175-181.

rules are the same described in the WOBANA program, GO-words in simple form (see example file FWORTE.DAT). The number of indicators for foreign words is dependent on the available memory (RAM). An indicator must not be longer than 80 characters.

The file REFO.SYL contains the patterns for the counting syllable algorithm, REFOD.SYL is German, REFOE.SYL for English. With the file REFO.SYL the algorithm for counting the syllables is controlled, and it can be adapted to other languages quite easily. The following rules apply:

- The longest patterns must occur at the beginning of the file.
- Up to 200 patterns are allowed.
- The maximum length of a pattern is 4 characters.
- Only capital letters are allowed.
- Within the patterns only ? as wild cards may be used (important for the English language). The * (asterisk) as a wild card character is not allowed.

In general these pattern are an enumeration of the diphtongs of a language. In languages with big differences between spoken and written language (e.g. English), whole syllables must be entered. The provided REFO*.SYL files show how that is done for German and English. The algorithm of syllable counting can be controlled by a protocol file.

6.14.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 15.05.1997 15:03	
program: REFO	
application: readability analysis	
input/output files	
name of system file	kontakt.itx
file of foreign words	fworte.dat
parameters	
process all text units	yes
protocol file for syllable counting	no
protocol file for foreign words	no
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

file of foreign words: For the calculation of TRI indicators for foreign words are counted, the indicators can be validated with this file, it contains the words being recognised as foreign words.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed (see chapter 5 on page 59).

protocol file for syllable counting: enter a file name if you want to validate the syllable counting algorithm, otherwise leave it empty.

protocol file for foreign words: enter a file name if you want to validate the foreign words recognising, otherwise leave it empty.

6.14.2 Information messages

```
INTEXT/586 Version 4.0 - 4/1997 - 23.05.1997 16:00
  program: REFO
    application: readability analysis
input file      kontakt.itx
file of special words fworte.dat
- I 01:      560 text units read
- I 02:     8829 strings read
- I 03:    57747 character read
- I 04:   15.766 strings/text unit
- I 05:    6.541 character/string
- I 06:  103.120 character/text unit
- I 07:    0.223 foreign words/text unit
- I 08:   11633 other strings read
syllables      0    1    2    3    4    5    6    7    8    9
words with 2721 2771 1905  870  420  101  32   7   2   0
- I 09:   20.773 syllables/text unit
- I 10:    1.318 syllables/string
- I 10:    125 foreign words read
- I 16:   68.696 value of MREI
- I 17:   52.001 value of NREI
- I 18:   27.016 value of MNREI
- I 21:  125.882 value of TRI
- I 22:   87.155 value of AVI
REFO started 16:00:43
REFO ended   16:01:00
REFO needed 17 seconds CPU-time
```

6.15 Personality structure analysis

The personality analysis by Mittenecker can be used to discriminate schizophrenic people from mentally healthy people. Schizophrenic people use significantly more repetitions of words or phrases than mentally healthy people. The PERSANA program counts this repetitions and writes them to an output file for further analysis.

Repetitions can be counted in two modes:

- exact 1:1 comparison of single words, with case folding enabled or disabled.
- a test, whether one word is part of another (infix position). This can result in a huge output file.

The format of the output file is as follows:

columns	content
1 - x	identifierws (variable long)
x+1-x+6	distance of the words (5 digit)
x+7 -	word

6.15.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 15.05.1997		15:03
program: PERSANA		
application: personality structure analysis		
input/output files		
name of system file		kontakt.itx
file of repetitions		kontakt.per
parameters		
process all text units		yes
case folding enabled		yes
include punctuation marks		no
F1 help files	F2 help parameters	F3 change
F4 start	F10 end	

name of system file: the name of the file where the system file is stored. The name may contain drive and/or directory specifications.

process all text units If you affirm this question, the complete text will be processed, otherwise the defined sample will be processed (see chapter 5 on page 59).

case folding enabled Letters can be treated as the same, if they are different only in their case (lower or upper case).

If this option is enabled, a test whether one word is part of another.

include punctuation marks Punctuation (all characters lower than "A") marks can be excluded from the comparison. Dependent on the genre of the text it is possible that rather huge parts of the output file consists of the distance of commas, if you answer with y(es).

6.15.2 Information messages

```
INTEXT/586 Version 4.0 - 2/1997 - 20.03.1997 16:27
      routine: PERSANA
      application: personality analysis
input file      kontakt.itx
output file     kontakt.per
case folding enabled
statistics:
- I 01:         6315 text units
- I 02:         1325 repeats by comparison
- I 03:         0 repeats in infix position
PERSANA start:  16:27:48
PERSANA end:    16:27:50
PERSANA needed 2 seconds CPU-time
```

6.16 Qualitative analyses of text

The purpose of text analyses in the social science is the collection of information, its ordering and analysis. Parts of the text are marked for further analysis.

In the context of quantitative text analysis techniques this means, that search patterns are grouped in categories, each with a numeric code. All search patterns are searched in every text unit. If a search pattern is found, the code of it is written to the output file.

In the context of qualitative data analysis the meaningful parts of a text are marked with codes. After that structures are searched, text segments are compared etc. Statistical analyses are done rather seldom, although they are possible.

A prerequisite of codings are search patterns. If INTEXT is used for a qualitative analysis, unique codes for marking parts of the text must be defined that can be used as search patterns. It is important that these search patterns are not ambiguous and that these cannot occur in the text.

An example: The number sign # can be used as a unique code that can be followed by a part of text of undefined length. The file KONTAKT.LAB contains the labels of the codes of the category system for the quantitative analysis of the text. These can be converted into a form suitable for qualitative analysis like this:

Code	category	qualitative code
1	kultureller Hintergrund	#Kultur
2	räumliche Mobilität	#RaumMobil
3	örtliche Gebundenheit	#örtlGeb
4	geistige Mobilität	#GeistMobil
5	Unkonventionalität	#Unkonv
6	Beruf / Residualkategorie	#BerufRest
7	akademische Berufe	#BerufAkad
8	hoher ökonomischer Status	#Status
10	politisch links	#links

Of course this example can be altered, but it is important that the # is followed by unique character combination. It is not absolutely necessary, that the # is followed by only one word, you can use more. They can be as long as you wish, but entering long codes takes more time to be written into the text and it is more likely that orthographical errors occur the longer these codes are.

INTEXT works with search patterns, so the marked parts of the text must be formulated as word root chains. The following example shows the technique (see also file QUAL.TXT):

```
Young man aged 30, 1,78cm, #single living alone #, wants to meet a girl
with
#body attributes a slim figure and long hair # for a long lasting
relationship.
```

In the example two parts of the text are marked, *living alone* for the category *single* and *a slim figure and long hair* for the category *body attributes*. If these parts of the text are to be analysed using a content analysis or a concordance, one can use the following word root chains as search patterns(see also file QUAL.DIC):

```
001 f '#single #'
002 f '#body attributes #'
```

7. Data management

7.1 Sorting

The ISM (INTEXT Sort/Merge) program supports sorting tasks. There is only the PC-version available. Nearly all relevant files can be sorted by alphabet or frequency, ascending or descending:

- strings, one string in each line
- strings with external variables (version 2.7)
- word list
- category system
- concordances
- search pattern in text unit

The following parameters can be defined:

- file format
- enable/disable case folding
- sorting criteria: ascending or descending
- sorting type: by alphabet or by frequency

Many languages have – other than English – characters with accents or diacritics e.g. ø, Ä, å), which are sorted behind the z in the ASCII character set. You can change the sort order in the ISM.DEF file, choose menu Edit with the submenu sort order table.

Umlauts and/or differences in upper-/lower case (case folding) may be enabled or disabled.

7.1.1 Parameters of the program

Which format does the input file have? There are five different sorting formats:

1. strings, one string in a line
2. SPLIT-file in long format (version 2.7)
3. word list
4. category system / search patterns
5. category labels
6. search patterns in text unit

Are differences in case sensitivity to be ignored? (y) Letters can be treated as the same, if they are different only in their case (lower or upper case).

What sort order is to be choosed? This question will only be asked if word lists are to be sorted. Sorting is possible by alphabet or by frequency, ascending or descending.

7.1.2 Information messages

```
INTEXT/PC Routine Sort/Merge 27.12.1990
Input file   kontakt.wb
Output file  kontakt.wba
sorting criteria: by frequency descending
- I 01:      6718 strings read
- I 02:      6718 strings written
- I 03:      61041 comparision done
- I 04:      11 passes
SORT start:  17:37:25
SORT end:    17:37:29
SORT needed 4 seconds CPU-time
```

7.1.3 Working with sort order tables

Many languages have – other than English – characters with accents or diacritics e.g. ø, Ä, å), which are sorted behind the z in the ASCII character set. You can change the sort order in the ISM.DEF file, choose menu Edit with the submenu sort order table. It can be edited with any editor or text processor that is able to write pure ASCII-files (stored unformatted with CR/LF). Sort order tables for several languages are supplied (ISM*.DEF).

The following sort order tables are provided:

language	file
English	ISM.DEF
German (dictionary)	ISM-D1.DEF
German (Telekom)	ISM-D2.DEF
French	ISM-F.DEF
Spanish	ISM-ESP.DEF
Cyrillic	ISM-KYR.DEF

You can use them by copying them into the file ISM.DEF. Example: `copy ism-d1.def ism.def`

Up to 60 characters (or combinations can be specified. Thus it is possible to translate other alphabets (cyrillic, greek, etc.) into the latin alphabet. For cyrillic texts it is possible to code single letters in multiple ones.

For German two sort modes (see DIN 5008) exist: in dictionaries umlauts are treated the same as their basic vowels, ä is treated like a. German Telecom (Postdienst Telekom) uses a different sort order for umlauts in their telephone directories: ä is treated like ae.

Each definition for a character is written in a new line. The following example is for German:

```

ä=a
ö=o
ü=u
Ä=A
Ö=O
Ü=U
ß=ss

```

7.2 Merging of vocabularies

If an error message says: no more RAM, than the vocabulary cannot be produced as a whole. Then there are two strategies how to achieve this:

- reducing it with STOP-words: word lists occupy a little less RAM than before, but word sequences, word permutations, and cross references use much less RAM with STOP-words. The price is that it takes up to 20 times longer.
- Generating of partial vocabularies: using samples one generates partial vocabularies and merges them to get a total vocabulary. The supervisor supports this function.

To generate a total vocabulary from several partial vocabularies proceeds as follows:

- generate alphabetically sorted partial vocabularies by using the sample definitions.
- copy the partial vocabularies into a file, e.g. with the copy command. Example:
If the word lists ONE.WB, TWO.WB, THREE.WB, and FOUR.WB are merged into the total word list TOTAL.WB, the files have to be copied with the following command: `copy ONE.WB+TWO.WB+THREE.WB+FOUR.WB TOTAL.WB`
- sort the file of all vocabularies by alphabet, e.g. the file TOTAL.WB. If the supervisor IS is used, the copy command and the calls of ISM and WOBANA are generated. Remember to use the same (or no) sort table for all (partial) vocabularies.
- merging the vocabularies: the file of the total vocabulary (example file TOTAL.WB) contains the strings following each other that occur in several partial vocabularies. The frequencies are added and written to the total vocabulary.

7.2.1 Parameters of the program

INTEXT/586 4.0 - 2/1997 - 06.02.1997 14:15					
program: WOBANA					
application: merge					
input/ouput file					
file word vocabulary				sport.wb	
file of merged vocabulary				sport.mwb	
F1	help files	F2	help parameters	F3	change
				F4	start
					F10 end

Only the files names have to be specified, there are no parameters.

7.2.2 Information messages

```
INTEXT/586 4.0 - 4/1997 - 21.04.1997 11:06
input file      md.wb
output file     md.mwb
```

```
statistics:
type of string      types      token   TTR
strings read
words               26289   177234  0.148
numbers             614     3598   0.171
other                59     25713  0.002

total               26962   206545  0.131
```

```
strings written:
words               26286   177234  0.148
numbers             614     3598   0.171
other                59     25713  0.002
total               26959   206545  0.131
```

```
differences (read - written strings):
type of string      types      %      token   %
words                3    0.011    0    0.000
numbers              0    0.000    0    0.000
other                 0    0.000    0    0.000
total                 3    0.011    0    0.000
```

```
WOBANA start: 11:06:06
WOBANA end:   11:06:09
WOBANA needed 3 seconds CPU-time
```

7.3 Reducing vocabularies

Another strategy of an explorative analysis of a vocabulary is to assume that strings that are rare or very frequent are irrelevant for one's own analysis, and do delete them from the vocabulary. Another criterion may be the length of a string. For example, a work file can be used for a later STOP word file that can be altered later. The following selection criteria can be used, also in combinations, only selection by frequency with cross reference is not possible:

- length, measured in number of characters
- frequency, both absolute values or in per cent (%) or per mille (‰), e.g. 3,4 ‰. All values are inclusive. If e.g. the minimum length is 3 and the maximum length is 10, then all strings with at least 3 and at most 10 characters are processed.
- occurrence in a STOP word file, e.g. ENGLISH.STP. Then all entries of the STOP word file are not written to the vocabulary. The file ENGLISH.STP – which is part of INTEXT – contains punctuation marks, articles, pronouns, and prepositions. The

entries in this file need not to be sorted by alphabet. Processing takes a lot more time then without STOP words.

All criteria can be combined. Length and frequency are specified by minimum and maximum values (inclusive values).

7.3.1 Parameters of the program

INTEXT/586 4.0 - 4/1997 - 30.05.1997 09:06	
program: WOBANA	
application: reduce with STOP-words	
input/output files	
file of word list	kontakt.wb
vocabulary without STOP-words	kontakt.wbs
selection criteria	
minimum length 1	minimum frequency 1
maximum length 80	maximum frequency 100000
name of STOP-word file	english.stp
STOP-words are whole words	yes
F1 help files	F2 help parameters
F3 change	F4 start
F10 end	

file of word list: the name of the file where the word list is stored. The name may contain drive and/or directory specifications.

vocabulary without STOP-words: the name of file where the vocabulary without the entries occurring in the STOP word file are written to. The name may contain drive and/or directory specifications.

minimum length: the minimum number of characters a string must have to be included in the vocabulary.

maximum length: the maximum number of characters a string may have to be included in the vocabulary.

minimum frequency: the minimum number of occurrences a string must have to be included in the vocabulary.

maximum frequency: the maximum number of characters a string may have to be included in the vocabulary.

name of STOP-word file: If you enter a valid file name, all strings that are in the STOP-word file will not be processed.

whole words: If you answer with yes, the strings to be excluded must be identical with the STOP word, if you answer with no, the STOP-word can be any part of the string. Case folding is enabled. Example: if the STOP-word "men" is not defined as a whole word, then also the strings "women" and "amen" are excluded.

7.3.2 Information messages

INTEXT/586 4.0 - 4/1997 - 30.05.1997 09:06

input file kontakt.wb
output file kontakt.wbs
STOP word file english.stp

statistics:

type of string	types	token	TTR
strings read:			
words	6095	41098	0.148
numbers	588	3806	0.154
other	35	17605	0.002
total	6718	62509	0.107

skipped strings in total

same	38	18474	0.002
total	38	18474	0.002

strings written:

words	6069	40192	0.151
numbers	588	3806	0.154
other	23	37	0.622
total	6680	44035	0.152

differences (read - written strings):

type of string	types	%	token	%
words	26	0.427	906	2.204
numbers	0	0.000	0	0.000
other	12	34.286	17568	99.790
total	38	0.566	18474	29.554

WOBANA start: 09:07:01

WOBANA end: 09:07:02

WOBANA needed 1 seconds CPU time

7.4 Convertings

7.4.1 Converting a reverse vocabulary into normal form

If a reverse vocabulary is printed, it has to be read from right to the left. With the converting function all strings are made readable, so that one can read them from left to right without losing the order they were sorted before. A right justified mode is possible, so that suffixes are placed on the same columns. The length is variable, but one has to have in mind how these lines are printed (or displayed) and which cpi (characters per inch, for printing) value fits best.

7.4.2 Converting a word list into a file of search patterns

During the construction of a category system it is useful that word lists can be formatted in such manner that search patterns can be edited easily. So a word list can be converted into a file of search patterns (only whole single words). The frequency has 3 digits, the parameter field contains a 'U' (for the upper-/lower case option, also known as case folding enabled). The word is embedded by single quotes. Using this file means that you have to change the frequency into a code and delete all strings that you don't need. Of course you can delete the leading and the trailing blanks in the search patterns.

7.4.3 Convert category system

The utility program T2I-DIC.EXE is invoked that converts a category system suitable for TEXTPACK into a file of search patterns for INTEXT. Vice versa is not possible because search patterns of INTEXT are much more powerful than the TEXTPACK ones. Parameters are not converted because they are consired by the conversion of the search pattern and because the C option has the same meaning in INTEXT and TEXTPACK. The name for the input and the output file are asked.

8. Working with code pages

In the last years code pages have been developed to gain independence from operating systems, so that texts containing characters that are not included in the English alphabet can be processed. This description is for MS-DOS systems (and compatibles) only.

These are mostly characters with diacritics or accents. It must be possible to enter them via the keyboard, to display them on the screen and to print them. To achieve it one can

- use drivers for screens, keyboards and printers. The configuration of the PC must be altered in the file CONFIG.SYS, and this is not a piece of cake.
- use systematic coding of all non displayable characters.

8.1 Using code pages

The most easiest way to achieve this is to use drivers for the keyboard, the screen, and the printer. The drivers use code pages. The selection of the code page is dependent on the language and its characters used in the text. The characters and their position within the code pages are shown on the next pages. MS-DOS provides drivers for screens and keyboards, but one has to take into account that not all combinations of screen and keyboard drivers will work.

Commands are to be inserted or changed in the files CONFIG.SYS and AUTOEXEC.BAT. The following example show how 6 code pages are prepared with the 850 code page as currently active code page:

```
COUNTRY=049,850,C:\DOS\COUNTRY.SYS
DEVICEHIGH =C:\DOS\DISPLAY.SYS CON=(EGA,,6)
```

Example for the AUTOEXEC.BAT file loading a German keyboard driver and 6 code-pages:

```
C:\DOS\KEYB GR,437,C:\DOS\KEYBOARD.SYS
```

nlsfunc

```
mode con cp prep=((437 850 852 860 865 863)c:\dos\ega.cpi)
```

```
mode con cp prep=(437)c:\dos\ega.cpi)
```

The following keyboard drivers can be activated with the keyb abbrev., Code /ID:IDnr command:

language/country	abbrev.	code	IDnr	country
english (international)		437, 850		061
Belgium	be	437, 850	120	032
Brasil	br	437, 850		055
Canada (french)	cf	863, 850	058	002
Czech	cz	852, 850		042
Denmark	dk	865, 850	159	045
Finland	su	437, 850	153	358
France	fr	437, 850	120, 189	033
Germany	gr	437, 850	129	049
Great Britain	uk	437, 850	166, 168	044
Hungary	hu	852, 850		036
Italy	it	437, 850	141, 142	039
Latin america	la	437, 850	171	003
Netherlands	nl	437, 850	143	031
Norway	no	865, 850	155	047
Poland	pl	852, 850		048
Portugal	po	860, 850	163	351
Slovakia	sl	852, 850		042
Spain	sp	437, 850	172	034
Sweden	sv	437, 850	153	046
Switzerland (german)	sg	437, 850	000	041
Switzerland (french)	sf	437, 850	150	041
USA	us	437, 850	103	001
Yugoslavia	yu	852, 850		038

Example: keyb gr, 437,c:keyboard.sys

Not every keyboard driver works together with every code page. You can change the code page with the change code page command CHCP, for example:

CHCP 852

→ This command changes the definitions of some keys of the keyboard, maybe you will not find some keys any more. The keyboard layout is documented in your MS-DOS manual.

The following table shows working combinations:

language/country	number
ANSI	1004
arabian	710
Canada (french)	863
cyrillic with accents	8660
cyrillic	866
cyrillic WordPerfect	899
Czech (Kamenicky)	895
Eastern Europe	852
Europe	333
greek	851
greek alternate	8510
hebrew	862
hungary (CWI-standard)	897
icelandic	861
multiple	850
norwegian	865
portugese	860
portugese/Brasoft	8600
portugese/Itautec	8601
turkish	857
USA	437

The tables on the following pages show which characters are included in which character tables.

character	437	850	852	860	863	865
à	133	133	—	133	133	133
á	160	160	160	160	—	—
â	131	131	131	131	131	131
Á	—	181	181	134	—	—
À	—	182	182	143	132	132
ä	132	132	132	—	—	—
Ä	142	142	142	—	—	—
å	134	134	—	—	—	—
Å	143	143	—	—	—	—
æ	145	145	—	—	—	—
Æ	146	146	—	—	—	—
Â	—	199	—	142	—	—
â	—	198	—	132	—	—
Ą	—	—	164	—	—	—
ą	—	—	165	—	—	—
Ă	—	—	198	—	—	—
ă	—	—	199	—	—	—
ḃ	—	231	—	—	—	—
B	—	232	—	—	—	—
ç	135	135	135	135	135	135
Ç	128	128	128	128	128	128
ć	—	—	134	—	—	—
Ć	—	—	143	—	—	—
č	—	—	159	—	—	—
Č	—	—	172	—	—	—
ḋ	—	—	208	—	—	—
Ḍ	—	—	209	—	—	—
ď	—	—	210	—	—	—
Ḑ	—	—	211	—	—	—
è	138	138	—	138	138	138
é	130	130	130	130	130	130
ê	136	136	—	136	136	136
ě	137	137	137	—	137	137
È	—	212	—	146	145	145
É	144	144	144	144	144	144
Ě	—	211	211	—	—	—
Ě	—	—	168	—	—	—
ę	—	—	169	—	—	—
Ě	—	—	183	—	—	—
ě	—	—	216	—	—	—

character	437	850	852	860	863	865
ì	141	141	—	141	141	141
ĩ	139	139	—	—	139	139
î	140	140	140	—	140	140
í	161	161	161	161	—	—
Í	—	214	214	139	—	—
Î	—	215	215	—	168	168
Ì	—	222	—	152	—	—
Ī	—	—	150	—	—	—
Ĭ	—	—	—	—	149	149
ł	—	—	136	—	—	—
Ł	—	—	157	—	—	—
Ł́	—	—	145	—	—	—
Ł̂	—	—	149	—	—	—
ñ	164	164	—	164	—	—
Ñ	165	165	—	165	—	—
ñ̂	—	—	229	—	—	—
Ñ̂	—	—	213	—	—	—
ń	—	—	228	—	—	—
Ń	—	—	227	—	—	—
Ń̂	—	—	213	—	—	—
ò	149	149	—	149	—	—
Ò	—	227	—	169	—	—
ó	162	162	162	162	162	162
Ó	—	—	224	159	—	—
ô	147	147	147	147	147	147
Ô	—	226	226	140	153	153
ö	148	148	148	—	—	—
Ö	153	153	153	—	—	—
õ	—	228	—	148	—	—
Õ	—	229	—	153	—	—
ö̂	—	—	139	—	—	—
Ö̂	—	—	138	—	—	—
ø	—	155	—	—	—	—
Ø	—	157	—	—	—	—
ř	—	—	234	—	—	—
Ř	—	—	232	—	—	—
ř̂	—	—	253	—	—	—
Ř̂	—	—	252	—	—	—
ś	—	—	152	—	—	—
Ś	—	—	151	—	—	—
š	—	—	173	—	—	—
Š	—	—	184	—	—	—

character	437	850	852	860	863	865
ĥ	—	—	156	—	—	—
Ť	—	—	155	—	—	—
ĥ	—	—	238	—	—	—
Ť	—	—	221	—	—	—
ù	151	151	—	151	151	151
ú	163	163	163	163	163	163
û	150	150	—	—	150	150
ü	129	129	—	129	129	129
Û	154	154	154	154	154	154
Ù	—	233	233	150	—	—
Û	—	234	—	—	158	158
ű	—	—	235	—	—	—
Ű	—	—	251	—	—	—
ÿ	152	152	—	—	—	—
ý	—	236	236	—	—	—
Ý	—	237	237	—	—	—
ž	—	—	171	—	—	—
Ž	—	—	141	—	—	—
ž	—	—	167	—	—	—
Ž	—	—	166	—	—	—
ž	—	—	190	—	—	—
Ž	—	—	189	—	—	—

8.2 Coding of non displayable characters

The second possibility is to code all non displayable characters. This is shown using \TeX , because with \TeX all accents and diacritics can be generated independent of hardware or operating system. This is an advantage especially if characters occur in the text that are not contained within a single code page. The sorting order is no problem, because the definition of multiple character letters in the file ISM.DEF is supported, for example:

```
\ "U = š \c{s} = ſ
```

So all accents and diacritics can be coded, and because \TeX contains drivers for screens and printers, professional outputs can be generated easily. A powerful advantage is that all characters can be used simultaneously – independent of the used code page – and that country specific keyboard drivers are not needed.

9. The INTEXT Result Manager

IRM is able to route the results to the screen, to the printer or to a file. The supervisor IS shows all files generated by INTEXT together with their file names. The following formats are supported:

- system files (also CODED-, REST- and NEG-files)
- vocabularies
- concordances in long and short format
- category systems
- output of a comparison of vocabularies
- search patterns in text units
- cross references

All parameters for printing are asked interactively, all values for the the page layout are required in mm (millimeter). Proportional spacing is not supported, but you can use different character and line densities. The margins (right, left, top, bottom) can be changed, also the paper length and paper width. IRM converts these specifications into the control sequences for the printer (nearly all printers can be adapted to work with IRM). Also single sheets can be used, also a centered heading on one single line, optional with page numbering (bottom or top, right, centered or left). The heading may also contain file name, date and time. One can use the following control sequences to achieve this:

- &d (date): the date in the form dd.mm.yyyy (European format)
- &t (time): time in the hh : mm : ss (24 hours)
- &f (file): file name

IRM also knows the paper length of the DIN formats A3 to A6, B3 to B6 and C3 to C6. Also the US-paper formats legal, exec, and letter are supported. The values for the page layout are stored in the PRINT.DEF file. They are defined as follows:

- left margin: distance between left paper edge and text
- right margin: distance between end of line and right paper edge
- line length: right margin - left margin
- character density: number of characters per inch (1 inch = 25.4 mm)
- line density: distance of lines between two lines (e.g. 1.0 or 0.8)
- top margin: distance between top paper edge and text or heading
- bottom margin: distance between text and lower paper edge

If a result is routed to a file, you are asked for the file name, whether it should be formatted for $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, the line length (in characters) and the distance between the lines.

10. Error messages

There are four different types of errors that can occur:

1. errors with hardware
2. errors with files
3. errors with questions of the menus
4. errors during the execution of a program

They are described on the following pages.

10.1 Errors with hardware

Errors with hardware can be problems dealing with the printer or the graphics card.

file PRINT.CFG is missing The file PRINT.CFG consists of control sequences for the printer and was not found in the current directory of the current drive. Drivers for popular printers (*.CFG) are part of INTEXT and can be activated by renaming. Details can be found in chapter 2.4.2 on page 12.

printer not connected or no power The printer is not switched on, or there is no connection between the printer and the computer, probably because the cable is not properly fixed or the cable is damaged.

printer not ready The printer was not activated. Details can be found in the manual for the printer. In some cases pressing the reset button helps.

no paper The printer has no paper, or the paper is not properly in the paper tray.

frequencies exceeded the printing width It was not possible to write the frequency of a string of a word list to the output file correctly. If for example the printing width is 3 digits, only frequencies up to 999 can be written correctly, if the printing width is 4 digits, frequencies up to 9999 can be written correctly. Statistics are computed correctly independent from the printing width.

Error: no VGA-card found Configuring a VGA-card with ITXINST or VGAINST no VGA-card could be found. If you are certain you have a VGA-card, the configuration must be performed by using the provided drivers or manually. Details can be found in chapter 2.8.1 on page 17. Probably the VGA-card is not 100% compatible. It is unlikely that extended text modes can be used.

no special mode for long concordances There is no extended text mode that provides 132 columns per line, long concordances cannot be displayed adequately using extended text modes with IRM.

no special mode for word lists There is no extended text mode that provides at least 55 columns per page. Word lists cannot be displayed adequately using extended text modes with IRM.

10.2 Errors with files

file doesn't exist In the menus you are asked for a file name until a valid file name of an existing file is entered. If you forgot the file name, you can execute a DOS command with `[!command]` or exit to DOS with `[DOS]`. (and back with EXIT to the supervisor IS). If neither of these features works, you can interrupt the program with CTRL-C and restart.

diskette/hard disk full writing file There is no more space available on the current disk or hard disk drive, the program is terminated. Before one can restart the program, some files have to be deleted (if you need them, copy them elsewhere before you do so). The name of the file involved is shown.

output file erroneous or

temp file erroneous or

file cannot be allocated An output file cannot be allocated, for three reasons: the file name is wrong (illegal characters or too long), or the maximum number of currently open files is too small. In these cases the `files=` parameter in the CONFIG.SYS file has to be enlarged, the minimum value is 20, values higher than 32 have no effect. The third reason – quite rare – is that the number of files in the current directory is exceeded. This can only happen if the root directory is the current directory, or with diskettes. In both cases the file must be written to another directory, or other files must be deleted (don't forget to copy them to another place before if these files are still needed).

DOS-error: file cannot be deleted or

file cannot be renamed This error message can occur in the ISM program. The reason may be that the maximum number of currently open files is too small. In this case the `files=` parameter in the CONFIG.SYS file has to be enlarged, the minimum value is 20. Another reason – quite rare – is that the number of files in the current directory is exceeded. This can only happen if the root directory is the current directory, or with diskettes. In both cases the file must be written to another directory, or other files

must be deleted (don't forget to copy them if these files are still needed). Another solution is to start ISM outside the supervisor IS.

file too big to be sorted This message occurs if the file size exceeds 64 KB. These files cannot be sorted by the MS-DOS SORT.EXE. One can use another sort program by configuring it using the submenu external programs.

10.3 Errors occurring with the menus

Not only with file names, also with the questions asked in menu mode user input may cause errors. This is often the case if characters instead of numbers are entered. Currently it is not possible to correct these errors, defaults are taken instead.

contains characters instead of numbers a character or a string instead of a number was entered. This message occurs if a system file is to be generated and the control sequence contains a character instead of a number as second character (after the dollar (\$)) using variable format.

is too long the data entered are too long.

must be a number, not a character a number was expected, but a character was found instead.

Warning: probably not enough space on drive The supervisor IS calculated that the space available on the current drive may not be sufficient for the selected task. Due to a rather conservative algorithm it is often possible to carry out the selected task. If in doubt, delete files you don't need (do not forget to copy them if you do need them).

first perform the content analysis, then the statistical analysis can be performed

A statistical analysis of content analysis results can only be performed if the content analysis was performed before.

program ISYS: DIM not between 1 and 10 Using fixed format the values of the position for the identifiers have to be specified. Only values between 1 and 10 are allowed, no other characters or strings.

ID... must be a number, not a character A number was expected, a character was found.

open round bracket is missing An open round bracket doesn't exist within a control sequence. The control sequence and the line number are shown. This error message is also generated if a blank follows the dollar symbol. Control sequences must not contain blanks.

close round bracket is missing A close round bracket doesn't exist within a control sequence. The control sequence and the line number are shown. This error message is also generated if a control sequence contains a blank. Control sequences must not contain blanks.

10.4 Errors during the execution of a program

wrong colours: The ANSI.SYS driver or an equivalent VGA-driver has not been included in the CONFIG.SYS file.

```

INTERRUPT 0DH, GENERAL PROTECTION FAULT possible illegal address
error code = 0000
eax = 00000000     esi = 0000EDA2     flags = 3246     ds = 0033
ebx = 00000000     edi = 0000F212     eip = 00003C0E   es = 0033
ecx = 00000001     ebp = DFFFEDC4     cs = 002B        fs = 0000
edx = 0002AF63     esp = DFFFED28     ss = 003B        gs = 0000

```

program crash with INTEXT/386, INTEXT/486, and INTEXT/586: The wrong storage managing program to manage EMS-memory is used, e.g. the Windows 3.1 version of EMM386.EXE and/or HIMEM.SYS. The DOS-versions of these programs must be used instead of the Windows versions. Another source of errors can be the data, especially raw data, the search patterns, and the category labels. Some errors are detected by the consistency check of the category system. In doubt consult the author of the program.

No more RAM The available memory is not sufficient for the selected analysis. If the PC-version is used, only the selection of text units or an upgrade to INTEXT/386 (or INTEXT/486 or INTEXT/586) helps. If these versions are used, extended memory (EMS or XMS) must be increased. If the available memory is already used, more memory must be installed. This error message can occur only with the following applications: content analysis, word lists, cross references, word combinations and word permutations especially, because the amount of RAM is dependent on the length and the number of text units. Details how to configure extended memory can be found in chapter 2.8.3 on page 20.

program ISYS: no number between 1 and 50 Using free format the character after the dollar sign must be between 1 and 50, but another character was found.

program ISYS: line is too long, termination Using fixed format the maximum line length of 5000 characters, using page, paragraph, or line mode the maximum length of 32500 characters was exceeded.

program ISYS: invalid control sequence The control sequence that starts with a dollar symbol \$ contains invalid characters. In most cases it is not followed by a number between 1 and 50 but by an open round bracket.

program ISYS: wrong control sequence, \$ is missing using free format the first character of the file is not a \$.

program ISYS: open round bracket is missing An open round bracket doesn't exist within a control sequence. The control sequence and the line number are shown. This error message is also generated if a blank follows the dollar symbol. Control sequences must not contain blanks.

program ISYS: close round bracket is missing A close round bracket doesn't exist within a control sequence. The control sequence and the line number are shown. This error message is also generated if a control sequence contains a blank. Control sequences must not contain blanks.

program ISYS: DIM-value is not between 1 and 10 the number of characters for an identifier must not exceed 10 characters. The value specified is not between 1 and 10 (inclusive), also it is possible that a character instead of a number was specified.

program ISYS: invalid characters in the text the text contains characters with values lower than 32 (ASCII character set). This message can point to data transmission or converting errors.

program ISM: error with ISM.DEF, = missing The file of the sort order table – ISM.DEF – contains characters or strings that have another sort order than the one of the character set used by the operating system (MS-DOS: ASCII, Windows: ANSI) and therefore must be treated different (in German the umlauts ä, ö, ü belong to these characters). In front of the = the character or string is found that is to be treated different, after the = is specified how it should be treated.

program REFO: more than 200 diphthongs are not allowed There are more than 200 lines in the REFO.SYL file, the REFO program is able only to process 200. If you need more, please inform the author of the program.

program SUWACO: inverted comma in wrong position, search pattern ...
In the file of search patterns a character is expected that also occurs at the end of a search pattern, commonly the inverted comma. Such a character was not found.

program SUWACO: number of categories too big for TAB-file In the file of search patterns the maximum value of the codes is too big, it cannot exceed 999.

program SUWACO: no number specified, try once more While coding interactive and changing the code, another character instead of a number was specified. After that error message the value can be entered again.

program SUWACO: more than 999 labels are not allowed The limit for the number of categories is 999. Check the file of search patterns, especially the first three columns.

program SUWACO: error in label code An error occurred while processing the file of the category labels. It is very likely that there is no blank between the code and the label, there must be at least one blank between the code and the label, and the code must start in column 1.

program SUWACO: text unit longer than specified This error should not occur with INTEXT 2.6, only in earlier versions. If a system file has not been generated by the ISYS program of INTEXT, this error occurs. You must use INTEXT to generate a system file.

program SUWACO: overlapping search patterns This message occurs during the coding of a content analysis. A part of a text – often a blank – is both the end of a search pattern and the beginning of another search pattern. This cannot be displayed

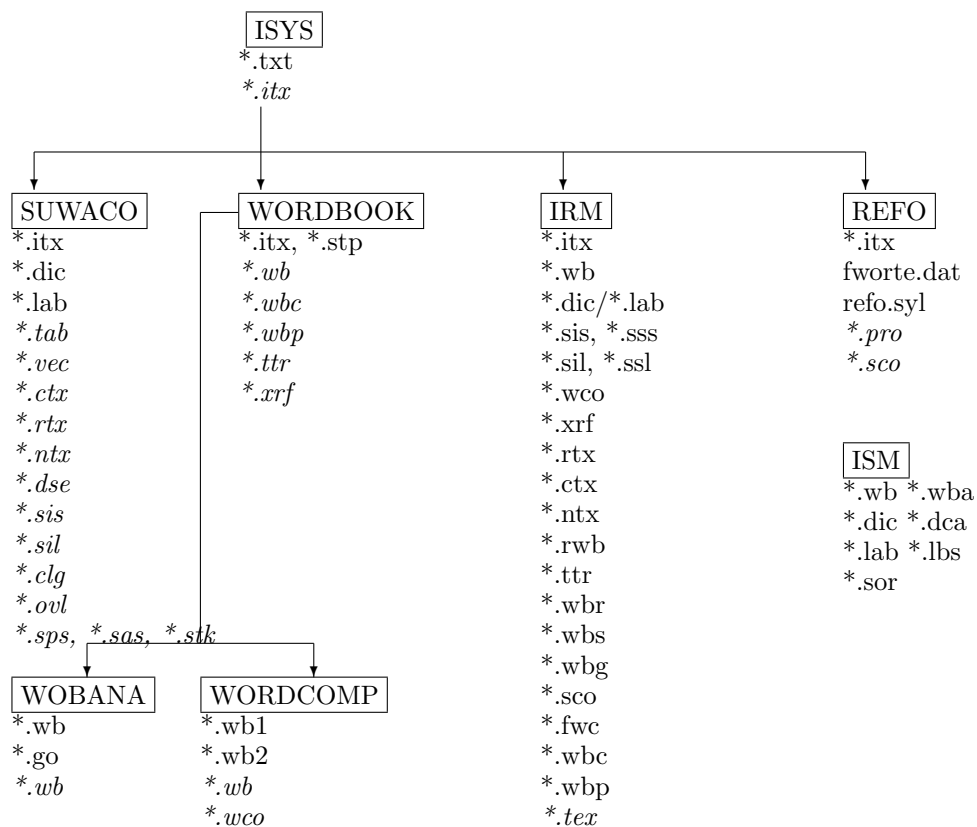
correctly during an interactive coding, while coding in vector mode only the code of the last search pattern will be stored. The counter of the codes are correct.

program SUWACO: overflow in . . .: while coding in vector mode more codes than the maximum number specified occurred. All other codes will not be stored. To obtain valid results, the coding should be repeated with an increased number.

program SUWACO: too many word roots The maximum number of word root chains is different, depending of the version. With the PC-version 200 word root chains are allowed, otherwise the maximum number is 1000.

program SUWACO: inconsistent category system, application aborted There is no category label for a code of a search pattern in the category system, or there are category labels for categories that were not used in the category system. The error messages contain the code, the label, and the search pattern.

11. The INTEXT programs



12. The structure of the INTEXT files

12.1 ITX-file: system file

The external variables are separated by tildes (~) and may consist up to 10 characters. Up to 50 external variables are possible. After the last external variable a vertical bar (|) follows, after that the number of words (5 digits) the length of the text (in characters), and a number sign (#). The text follows (maximum 100000 characters).

12.2 DIC file: search patterns

1	-	3	code (optional)
4	-	6	parameter field
7	-	200	search pattern

12.3 WB? file: word list, word combinations, word permutations

1	-	6	frequency of the string
		7	free
8	-	1000	string

12.4 XRF file: cross references

			1. line
1	-	80	word
			following lines
			external variables, separated by tildes (~)

12.5 VEC file: sequence of codes

1	-	x	External variables: x=number*10
x+1	-	x+5	strings in the text unit, 5 digits
x+6	-	x+10	codes in the text unit, 5 digits
x+11	-	x+14	counter 1. category, 3 digits
x+15	-	x+17	counter 2. category, 3 digits
x+18	-	x+20	counter 3. category, 3 digits

12.6 TAB file: code counter

1	-	x	External variables: x=number*10
x+1	-	x+5	strings in the text unit, 5 digits
x+6	-	x+10	codes in the text unit, 5 digits
x+11	-	x+14	1. code, 3 digits
x+15	-	x+17	2. code, 3 digits
x+18	-	x+20	3. code, 3 digits

12.7 sss/ssl/sis/sil file: concordances

The following example assumes 131 characters (specified in SUWACO) in one line.

1	-	3	code
		4	free
20	-	70	text before the search pattern
71	-	132	search pattern and following text

12.8 WCP file: short word comparison

1	-	9	frequency of the string in the 1. file
10	-	18	frequency of the string in the 2. file
19	-	27	difference of the frequencies
		28	free
29	-	108	word

12.9 WCP file: long word comparison

1	-	7	frequency of the string in the 1. file
8	-	46	string in the 1. file
47	-	53	difference of the frequencies
54	-	60	frequency of the string in the 2. file
61	-	99	string in the 2. file

12.10 TTR-file long: TTR-dynamics

1	-	40	token
41	-	49	cumulated value of the types
50	-	58	cumulated value of the token
59	-	64	TTR-value

12.11 TTR file short: TTR-dynamics

1	-	9	cumulated value of the types
10	-	18	cumulated value of the token
19	-	24	TTR-value

13. Bibliography

Anonymous (1989): A Short Guide to the General Inquirer. In: Bulletin de Méthodologie Sociologique 24, p. 6-8.

Ballstaedt, Steffen-Peter; Heinz Mandl; Wolfgang Schnotz; Sigmar-Olaf Tergan (1981): Texte verstehen, Texte gestalten. München, Wien, Baltimore.

Bausch, Karl Heinz (1973): Zur Umschrift gesprochener Hochsprache. In: IDS, Gesprochene Sprache. Mannheim.

Bierschenk, Bernhard (1977): A Computer-Based Content Analysis of Interview Texts: Numeric Description and Multivariate Analysis. In: Didakometry 53, p. 42.

Bierschenk, Bernhard (1978): Content Analysis as Research Method. In: Kompendieserien 25, p. 93.

Bierschenk, Inger (1977): Computer-Based Content Analysis: Coding Manual. In: Pedagogisk Dokumentation 52, p. 113.

Boot, N.N.M. (1978): Ambiguity and Automated Content Analysis. In: MDN, Methoden en Data Nieuwsbrief van de Sociaal-Wetenschappelijke Sectie van de VVS, 3/1, p. 117-137.

Boot, M.N.M. (1979): Homographie, ein Beitrag zur automatischen Wortklassenzuweisung in der Computerlinguistik. Utrecht.

Bos, Wilfried; Christian Tarnai (1989): Angewandte Inhaltsanalyse in Empirischer Pädagogik und Psychologie. Münster.

Chotlos, John W. (1944): A Statistical and Comparative Analysis of Individual Written Language Samples. Psych. Monographs 56/ Nr. 2, p. 75-111.

Clubb, Jerome M; Erwin K. Scheuch (eds. 1980): Historical and Process-Produced Data. Stuttgart.

Cuilenberg, Jan J. van; Jan Kleinnijenhuis; Jan A. de Ridder (1988): Artificial Intelligence and Content Analysis: Problems of and Strategies for Computer Text Analysis. In: Quality and Quantity 22/1, p. 65-97

Dasgupta, Atis K. (1975): A Note on Content Analysis. In: Sociological Bulletin 24/1, p. 87-94.

Deichsel, Alexander (1975): Elektronische Inhaltsanalyse. Zur quantitativen Beobachtung sprachlichen Handelns. Berlin.

DeWeese III, Carroll (1976): Computer content analysis of printed media. A feasibility study. In: Public Opinion Quarterly 40, p. 92-100.

DeWeese III, Carroll (1977): Computer content analysis of 'Day Old' Newspapers: A feasibility study. In: Public Opinion Quarterly 41, p. 91-94.

Dohrendorf, Rüdiger (1990): Zum publizistischen Profil der "Frankfurter Allgemeinen Zeitung". Computerunterstützte Inhaltsanalyse von Kommentaren der FAZ. Frankfurt/M, Bern, New York, Paris.

Drewek, Raimund (1985): LDVLIB – Textanalyse mit System. In: Lehmacher, Walter; Allmut Hörmann (eds.): Statistik-Software. 3. Konferenz über die wissenschaftliche Anwendung von Statistik-Software. Stuttgart, p. 283-296.

Fan, David P. (1988): Predictions of Public Opinion from the Mass Media: Computer Content Analysis and Mathematical Modeling. (s.l.): Greenwood Press.

Faulmann, Carl (1880): Das Buch der Schrift. Wien, Reprint Nördlingen 1985.

Fischer, Peter Michael (1982): Inhaltsanalytische Auswertung von Verbaldaten. In: Huber, Günter L.; Heinz Mandl: Verbale Daten. Weinheim and Basel, p. 179-196.

Fonnes I: (1974): TEXT: A General Program Package for Text Processing. In: Contributed Papers: ISSC-SCSSD Workshop on Content Analysis in the Social Sciences, Pisa CNUCE, August 1974, p. 77-83.

Franzosi, Roberto (1990): Computer-Assisted Coding of Textual Data. An Application to Semantic Grammars. In: Sociological Methods and Research 19/2, p. 225-257.

Frisbee, B.; S. Sudman (1968): The Use of Computers on Coding Free Responses. In: Public Opinion Quarterly 32, p. 216-232.

Früh, Werner (1984): Konventionelle und maschinelle Inhaltsanalyse im Vergleich: Zur Validierung computerunterstützter Bewertungsanalysen. In: Klingemann, Hans-Dieter (eds.): Computerunterstützte Inhaltsanalyse in der empirischen Sozialforschung. Frankfurt/Main, p. 35-53.

Frow, John (1989): Formal Method in Discourse Analysis. In: Journal of Pragmatics 13/3, p. 333-341.

Giegler, Helmut (1991): Zur computerunterstützten Analyse sozialwissenschaftlicher Textdaten. Quantitative und qualitative Strategien. In: Hoffmeier-Zlotnik, Jürgen (ed.): Analyse qualitativer sozialwissenschaftlicher Daten. Opladen, p. 335-388.

Heinrich, Horst-Alfred (1996): Traditional versus computer aided content analysis. A comparison between codings done by raters as well as by INTEXT. In: Faulbaum, Frank; Wolfgang Bandilla (eds.): SoftStat '95. Advances in statistical software 5. The 8th Conference on the Scientific Use of Statistical Software. March 26-30, 1995 Heidelberg. Stuttgart, p. 327-333.

Heinrich: Horst-Alfred (1996): Generationsbedingte zeithistorische Erinnerung in Deutschland. Ergebnisdokumentation einer computergestützten Inhaltsanalyse mit INTEXT. (= Nationale Identität. Arbeitsberichte aus dem DFG-Projekt "Nationale Identität der Deutschen". Messung und Erklärung der Veränderungsprozesse in Ost und West. Nr. 10), Mannheim.

Heinrich, Horst-Alfred (1996): Zeithistorische Ereignisse als Kristallisationspunkte von Generationen. Replikation eines Messinstrumentes. In: ZUMA-Nachrichten 39, p. 69-94.

Herdans, Gustav (1964): Quantitative Linguistics. London.

Johnson, Wendell (1944): Studies in Language Behaviour. Psych. Monographs 56/ Nr. 2

Klein, Harald (1988): INTEXT - ein Programmsystem zur computerunterstützten Inhaltsanalyse. In: Faulbaum, Frank; Hans-Martin Uehlinger (eds.): Fortschritte der Statistik-Software 1. Stuttgart, p. 574-581.

Klein, Harald (1990): New Possibilities and Developments of Text Analysis with INTEXT/PC. In: Faulbaum, Frank, Reinhold Haux; Karl-Heinz Jöckel (eds.): Fortschritte der Statistik-Software 2. Stuttgart, p. 487-494.

Klein, Harald (1990): INTEXT/PC – A Program Package for the Analysis of Texts. In: Universität Siegen (ed.): ALLC – ACH 90 The New Medium. Book of Abstracts & Conference Guide, p. 133-136.

Klein, Harald (1991): INTEXT/PC – A Program Package for the Analysis of Texts in the Humanities and Social Sciences. In: Literary and Linguistic Computing 6/2, p. 108-111.

Klein, Harald (1992): Validity Problems and their Solutions in Computer-Aided Content Analysis with INTEXT/PC and Other New Features. In: Faulbaum, Frank; Reinhold Haux; Karl-Heinz Jöckel (eds.): Advances in Statistical Software 3. Stuttgart, p. 483-388.

Klein, Harald (1993): INTEXT/PC – A Program Package for the Analysis of Texts. In: Steyer, Rolf, u.a (eds.): Proceedings of the 7th European Meeting of the Psychometric Society in Trier, Stuttgart, p. 219-221.

Klein, Harald (1993): INTEXT – a program system for the analysis of texts. In: Hřebiček, Luděk; Gabriel Altmann (eds.): *Quantitative Text Analysis*, p. 297-307. Trier: Wissenschaftlicher Verlag.

Klein, Harald; Helmut Giegler (1994): *Correspondence Analysis of Text Data with INTEXT/PC*. In: Greenacre, Michael; Jörg Blasius (eds.): *Correspondence Analysis in the Social Sciences*, p. 283-301. London: Academic Press.

Klein, Harald (1996): *Computerunterstützte Inhaltsanalyse mit INTEXT – dargestellt am Vergleich von Nachrichtenfaktoren des Fernsehens*. Münster.

Kleinen, Günter (1994): *Die psychologische Wirklichkeit der Musik. Wahrnehmung und Deutung im Alltag*. Kassel: Gustav Bosse Verlag.

Klingemann, Hans-Dieter (ed. 1980): *Computerunterstützte Inhaltsanalyse in der empirischen Sozialforschung. Anleitung zum praktischen Gebrauch*. Frankfurt am Main.

Klingemann, Hans-Dieter; Klaus Schönbach; Bernd Wegener (1978): *Nachrichtenwerte und computerunterstützte Inhaltsanalyse*. In: *ZUMA-Nachrichten* 2, p. 3-11.

Klingemann, Hans-Dieter; Peter Ph. Mohler (1979): *Computerunterstützte Inhaltsanalyse (CUI) bei offenen Fragen*. In: *ZUMA-Nachrichten* 4, p. 3-19.

Klingemann, Hans-Dieter; Peter Ph. Mohler (1980): *Deutsche Diktionäre für computerunterstützte Inhaltsanalyse (1)*. In: *ZUMA-Nachrichten* 6, p. 53-57

Kramer-Santel, Claudia (1995): *Die Darstellung des Umweltproblems in der Presse unter besonderer Berücksichtigung anreizkonformer Instrumente*. Dissertation, Münster.

Kuckartz, Udo (1988): *Computer und verbale Daten*. Zürich.

Laffal, Julius (1990): *A Concept Dictionary of English with Computer Programs for Content Analysis*. Essex, Ct.

Lavigne, Gilles; Joelles Martin; Elise Nantel (1989): *L'analyse de contenu assistée par ordinateur: L'option LIAO*. In: *La Revue Canadienne de Sociologie et d'Anthropologie*, 26/4, p. 596-616.

Lenders, Winfried; Gerd Willè (1986): *Linguistische Datenverarbeitung. Ein Lehrbuch*. Opladen.

Mandelbrot, Benoit (1961): *On the Theory of Word Frequencies and on Related Markovian Models of Discourse*. In: Roman Jakobson (eds.): *The Structure of Language*. Providence, p. 190-219.

McGee, Victor E. (1986): *The OWL: Software Support for a Model of Argumentation*. In: *Behavior Research Methods, Instruments & Computers* 18/2, p. 108-117.

McTavish, Donald G.; Ellen B. Pirro (1990): Contextual Content Analysis. In: *Quality and Quantity* 24/3, p. 245-265.

Messelken, H. (1989): Computerunterstützte Textanalyse. In: *Historical Social Research* 14/4, p. 86-93.

Mochmann, Ekkehard (1974): Automatisierte Textverarbeitung. In: Koolwijk, Jürgen van; Maria Wieken-Mayser (eds.): *Techniken der empirischen Sozialforschung*. 3. vol: Erhebungsmethoden. Beobachtungen und Analyse von Kommunikation. München, p. 192-202.

Mochmann, Ekkehard (1985): Inhaltsanalyse in den Sozialwissenschaften. In: *Sprache und Datenverarbeitung* 9/2, p. 5-10.

Mohler, Peter Ph. (1980): Deutsche Diktionäre für computerunterstützte Inhaltsanalyse (2). In: *ZUMA-Nachrichten* 7, p. 42-44.

Mohler, Peter Ph. (1981): Deutsche Diktionäre für computerunterstützte Inhaltsanalyse (3) In: *ZUMA-Nachrichten* 8, p. 51-53.

Mohler, Peter Ph. (1985): Computerunterstützte Inhaltsanalyse: Zwischen Algorithmen und Mythen. In: *Sprache und Datenverarbeitung* 9/2, p. 11-14.

Mohler, Peter Ph.; Cornelia Züll; Alfons Geis (1989): Die Zukunft der computerunterstützten Inhaltsanalyse (cui). In: *ZUMA-Nachrichten* 25, p. 39-46.

Mohler, Peter Ph. (1989): Die linguistischen Leistungen der computerunterstützten Inhaltsanalyse. In: Batori, Istvan; Wilfried Lenders; W. Putschke (eds.): *Computerlinguistik: Ein Internationales Handbuch der Computerunterstützten Sprachforschung und ihrer Anwendungen*. Berlin.

Mohler, Peter Ph.; Katja Frehsen; Ute Hauck (1989): CUI: Computerunterstützte Inhaltsanalyse. Grundzüge und Auswahlbibliographie zu neueren Anwendungen. Mannheim: ZUMA-Arbeitsbericht, Nr. 89/09.

Muskens, George (1985): Mathematical Analysis of Content. In: *Quality and Quantity* 19/1, p. 99-103.

Nath, Detlev W. (1979): COFTA – Compiler für Textanalysen (Einführung). St. Augustin.

Richardson, M.G. (1979): Verzeichnis Deutscher Diktionäre für computerunterstützte Inhaltsanalyse. In: *ZUMA-Nachrichten* 4, p. 20-22.

Roberts, Carl W. (1989): Other than Counting Words: A Linguistic Approach to Content Analysis. In: *Social Forces* 68/1, p. 147-177.

Roberts, Carl W.; Roel Popping (1993): Computer-supported Content Analysis: Some Recent Developments. In: *Social Science Computer Review* 11, p. 283-291.

Salton, G.; C.S. Yang; C.T. Yu (1975): A Theory of Term Importance in Automatic Text Analysis. In: *Journal of the American Society for Information Science* 26/1, p. 33-44.

Schnurr, Paula P.; Stanley D. Rosenberg; Thomas E. Oxman (1992): Comparison of TAT and Free Speech Techniques for Eliciting Source Material in Computerized Content Analysis. In: *Journal of Personality Assessment* 58/2, p. 311-325.

Schönbach, Klaus (1979): Elektronische Inhaltsanalyse in der Publizistikwissenschaft. In: *Publizistik* 24, p. 449-457.

Schönbach, Klaus (1982): "The Issues of the Seventies". Elektronische Inhaltsanalyse und die langfristige Beobachtung von Agenda-Setting-Wirkungen der Massenmedien. In: *Publizistik* 27, p. 129-139.

Sedelow, Walter A.; Sally Y. Sedelow (1978): Formalized Historiography, the Structure of Scientific and Literary Texts. Part 1: Some Issues Posed by Computational Methodology. In: *Journal of the History of the Behavioral Sciences* 14/3, p. 247-263

Sells, P. (1985): *Lectures on Contemporary Syntactic Theories*. Stanford.

Singh, Jaspal (1985): Content Analysis. In: *Guru Nanak Journal of Sociology* 6/1, p. 37-44.

Smith, Robert B.; Peter K. Manning (1982): *A Handbook of Social Science Methods. Volume 2: Qualitative Methods*. Cambridge

Spack, Jones K.; M. Kay (1976): *Linguistik und Informationswissenschaft*. München.

Stone, Philip J.: (1962): The General Inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information. In: *Behavioral Science* 7, p. 484-494.

Stone, Philip J. and Cambridge Computer Associates Inc. (1968): *User's Manual for the General Inquirer*. Cambridge, Mass..

Stone, Philip J.: (1969): Improved Quality of Content Analysis Categories: Computerized Disambiguation Rules for High-Frequency English Words. In: Gerbner, G. et al. (eds.): *The Analysis of Communication Content*: New York, p. 199-221.

Tiemann, Rainer (1973): *Algorithmisierte Inhaltsanalyse: Prozeduren zur Inhaltsanalyse verbaler Verhaltensweisen*. Hamburg.

Trappes-Lomax, H.R. (1974): *A Computer Based System for Content Analysis, a Review of the Edinburgh 'New Tagger' Version of the General Inquirer*. Edinburgh.

Trauth, Michael (1992): Quantifizierende Textanalyse. Mit der Hilfe des Computers auf der Suche nach dem anonymen Autor. In: *Historische Sozialforschung* 17/1, p. 133-141.

Weber, Heinz-Josef (1976): Automatische Lemmatisierung. In: Linguistische Berichte 44, p. 30-47.

Weber, Robert P. (1983): Measurement Models for Content Analysis In: Quality and Quantity 17/2, p. 127-149.

Weber, Robert P. (1984): Computer-Aided Content Analysis: A Short Primer. In: Qualitative Sociology 7/1-2, p. 126-147.

Weber, Robert P. (1986): Correlational Models of Content: Reply to Muskens In: Quality and Quantity 20, p. 2-3, 273-275.

Weber, Robert P. (1990): Basic Content Analysis. 2. ed., Newbury Park.

Wickmann, Dieter (1969): Eine mathematisch-statistische Methode zur Untersuchung der Verfasserfrage literarischer Texte. Durchgeführt am Beispiel der "Nachtwachen" von Bonaventura mit Hilfe der Wortartübergänge. Köln/Opladen (Forschungsberichte des Landes NRW Nr. 2019)

Wilde Kelly, Ann; A.M. Sine (1990): Language as Research Data: Application of Computer Content Analysis in Nursing Research. In: Advances in Nursing Science 12/3, p. 32-40.

Wood, Michael (1980): Alternatives and Options in Computer Content Analysis. In: Social Science Research 9/3, p. 273-286.

Woodrun, Eric (1984): Mainstreaming Content Analysis in Social Sciences: Methodological Advantages, Obstacles and Solutions. In: Social Science Research 13/1, p. 1-19.

Züll, Cornelia; Robert P. Weber; Peter Ph. Mohler (1989): Computer-aided Text Classification for the Social Sciences: The General Inquirer III. Mannheim.

Züll, Cornelia; Peter Ph. Mohler; Alfons Geis (1991): Computerunterstützte Inhaltsanalyse mit TEXTPACK PC Release 4.0 für IBM XT/AT und Kompatible unter MS/DOS ab Version 3.0. Stuttgart.

Züll, Cornelia; Peter Ph. Mohler (eds.) (1992): Textanalyse. Anwendungen der computerunterstützten Inhaltsanalyse. Opladen.

14. Glossary

The glossary explains the technical terms used in this manual.

ambiguity This problem occurs while defining search patterns for a category system (dictionary). Because search entries have to be defined unique, ambiguity must not occur. Example: pot. This can mean the same as a cup, but it can also mean a certain drug. The search pattern ' pot ' is ambiguous. It makes sense that you examine the context by doing a concordance of the text unit.

analysis unit see coding unit.

blank Another word for space. A word is formed by all characters between two blanks (or other delimiters like start or end of a line).

case folding Enabling case folding means that strings (mostly words) that are only different because they differ in lower/upper case letters are treated as the same by some INTEXT programs. Disabling case folding means that all differences matter, also the one that are based on differences in upper/lower case. For example: That and that are treated as one word if case folding is enabled and as two words if case folding is disabled.

category Operationalisation of a theoretical construct with one or more search patterns (see there).

category system a group of several categories. Every category consists of at least one search pattern.

character string all characters between two blanks (see there), usually a word.

coding unit The coding unit (see program SUWACO) is the definition of a case. A new coding unit starts with every new text unit. In former versions of INTEXT this was different, coding results could be aggregated. Aggregation with version 3.0 can only be performed with statistics software (e.g. Aggregate within SPSS).

concordance Search patterns in their context. This is an analysis that shows search patterns and their context in one line (similar to KWICs). The search patterns are in the center of a line, the rest consists of the context before and after the search pattern. In INTEXT the length of the line is variable, and the results can be formatted in two ways: as short and long concordances.

cross reference A lists of all positions of a string where it occurs. A cross reference consists of all external variables and their positions within the text nmit.

data generation The goal of the data generation is to make a text readable by a computer (machine readability). Currently that is only possible using an editor. In INTEXT the goal is to generate a system file (see there).

default Each parameter that can be changed by the user has a value that is taken if the user doesn't specify the parameter, this is called the default, e.g. file names have default names derived from the name of the project.

dictionary another term for category system. A dictionary consists of all search patterns that form the categories. Sometimes the term dictionary is also used in the sense of a word list.

digit all strings where the first characters is a digit (0-9).

external variable These variables represent variables of a text. They must be specified by the user, up to 50 external variables are possible, at least one is required.

file A form how to organise data. A file consists of logical records, each record consists of at least one variable. Logical records of a file of text units (the INTEXT system file) consist of the three external variables, the number of words, the numbers of characters and the text. Each file has its own structure, the details are described in chapter *Structure of the files*.

floating text Text in the format of a floating text is organised in a file that consists of text units as a logical record. This is the format a system file is organised. Another form of organising text is the vertical text format where a logical records consists of the external variables and one word.

format Every file has a format that describes how the logical records look like (where which variables are to be found). The formats are described in detail in the chapter *Structure of the files*. INTEXT has two raw data formats (fixed and variable) and a system file.

foreign word Readability analyses that use the TRI formula (for political comments in German newspapers) take foreign words into account, that means these words are used in German but have a different origin (e.g. Greek or Latin). Foreign words are also called special words (see there).

homonym A string that has more than one meaning. In a content analysis homonyms have to be disambiguated (see ambiguity). Example: pot. Meaning: cup or drug.

hyphenation The hyphenation of words in a raw text is not allowed. All hyphenated words have to be eliminated before the system file is generated.

infix A string (see there) that may occur in any position within a word (see there). If an infix occurs in the beginning of a string, it's called prefix (see there), if it occurs at the end of a string, it's called suffix (see there). In a strict sense an infix may not occur at the beginning or end of a string).

list of uncoded words is a word list (see there) of all strings (see there) that are not coded by a content analysis using the search pattern of a category system. Basis of a list of uncoded words are a word list and a category system (see there) that is sorted ascending by alphabet.

numeral a number written as a word (e.g.: one, eleven).

output file Many programs write their results into output files that can be processed by other programs (see also file). The output file of the ISYS program is the input file of many other programs (e.g. WORDBOOK, SUWACO). The formats (see there) of all files are described in the chapter *Structure of the files*.

prefix A string (see there) that is in the beginning of a word (see there). A prefix is a special form of an infix (see there). In content analysis that can be a single letter (or another character).

raw text machine readable form of a text that can be processed without editing or converting by the ISYS program of INTEXT, so that a system file (see there) can be generated. The raw text must have specific formats, see the chapter of data preparation for details.

reverse word list word list (see there) where the words are listed in reverse order (the first character becomes the last, the last character becomes the first). Example: `small` becomes `llams`.

sequence number Raw texts in fixed format (see there) must contain a sequence number because a text unit often does not fit in an input line (see ISYS program).

special characters all characters that neither start with a letter or a number. These are e.g. punctuation marks or other characters of the characters set (IBM EBCDIC, PCs ASCII, Windows ANSI).

special word see foreign word.

STOP-word A word list (see there) contains all types (see there) of a text. Many of them are not useful for the definition of search patterns. Using a STOP-word file these can be deleted from a word list. Such a file contains articles, pronouns, prepositions and conjunctions.

search pattern at least one operationalisation of a category (see there). There are two types of search patterns in INTEXT:

1. strings (words, part of words or sequences of words)
2. word root chains)

string a set of characters that is delimited by a blank in the beginning and the end (or other delimiters).

suffix that part of a string (see there) that forms the end of a string (see there). Search patterns can be defined as suffixes.

system file A file of text units (see there) that is the basic file for all forms of text analyses. They consist of external variables and the text, the latter is stored with variable length. A system file consists of at least one text unit (see there).

text unit A text unit is the unit of all further analyses and dependent what is to be researched. In readability analysis a text unit must be a sentence, in coding open ended questions a text unit is one answer to one open ended question. More details are described in *Preparation of the text*.

token another term for a string (see there) in a text, used in linguistics.

truncate A string can be truncated if it exceeds the maximum length of 80 characters in the following applications: cross references, sorting with IRM (if a sort order table is enabled, the maximum length of a string is 38 characters), some forms of output of comparisons of word lists.

TTR Type-Token-Ratio. The ratio between all different strings (types, see there) and the sum of all strings (token, see there). The value of the TTR is between 0 and 1; the higher it is, the more heterogen is the vocabulary of the text. A value of 0 indicates an empty input file, a value of 1 means, that each word occurs only once.

type the sum of different strings (see there) in a text.

variable format one of the many input formats of raw text (see there) that works with control sequences that start with \$. It is best used if you have to type in the text yourself.

vertical text The logical record of a text consists of a word together with its external variables. The opposite is called floating text (see there), each logical record consists of a text unit (see there).

word A word within a text unit are all characters, that are between two blanks (or another delimiter like start or end of a line). The more precise expression is string (see there), although most strings are words.

word list a list of all types (see there) together with their frequency. Sometimes the term frequency table is also used.

word length the number of characters in a string.

word root A string (see there), that can be part of another string. Word roots can be in prefix, infix or suffix position (see there).

word root chain several word roots that must occur within one text unit. Up to 6 word roots can be in a word root chain. These can be searched within a text unit in three different modes that vary the order and the distance of the word roots how they must occur in the text (see chapter about the search patterns).

15. Index

Index

- 386MAX, 10, 20
- accents, 45, 129
- Altmann, Gabriel, 29, 91
- ambiguity, 28–30, 45, 95, 103, 106, 157, 158
- ambiguous search pattern, 106
- analysis
 - personality structure, 118
 - readability, 31
 - statistical, 31, 50
- ANSI, 159
- ANSI.SYS, 11, 17, 22
- ASCII, 121, 159
- AUTOEXEC.BAT, 10–12, 129
- backup copy, 11
- beep, 13, 36
- blank, 157
- BUFFER
 - CONFIG.SYS, 21
- case folding, 23, 28, 29, 45, 48, 63, 79, 89, 91, 94, 103, 118, 121, 128, 157
 - GO words, 89
 - program ISM, 122
 - program PERSANA, 119
 - program WOBANA, 89
 - program WORDBOOK, 64, 67, 72, 76, 80, 91
 - search pattern, 31
- category, 103, 120, 157
- category building, 125
- category label, 25, 29, 103, 105, 110, 120
 - length, 103
- category system, 25, 28–30, 63, 94, 97, 99, 101, 103, 108, 110, 119, 121, 128, 135, 157, 158
- character set, 45, 129
- character spacing, 19
- character string, 157
- characters, 160
 - non displayable, 134
 - truncate, 47
- cluster analysis, 31
- code, 94, 105
 - counter, 108
 - order, 108
 - vector file, 108
- code page, 45, 129
- coded text units, 94
- coding, 45, 106, 108
 - interactive, 106
- coding control, 94, 107
- coding result, 105
- coding unit, 105, 157
- colours, 17, 22, 140
- comparision
 - complete, 84
 - vocabularies, 83, 135
- compatibility, 21
- ConClus, 13, 31, 36, 39, 105, 107
- concordance, 18, 28, 30, 32, 59, 98, 99, 120, 121, 135, 138, 139, 146, 157
- CONFIG.SYS, 10, 11, 17, 20–22, 129, 138, 140
- configuration, 11, 13, 36, 137
 - VGA-adaptor, 17
- content analysis, 9, 13, 23, 28, 30, 31, 43, 47, 59, 63, 94, 103, 157
 - qualitative, 45, 119
- continuous paper, 19
- control sequence, 9, 19, 42, 44–48, 52, 137, 140, 160
 - printer, 15
- convert, 33, 128
- counting unit, 157
- COUNTRY, 13
- crash, 140
- cross reference, 12, 29, 42, 59, 67, 79, 82, 135, 145, 157
 - reverse, 67
- cursor keys, 38
- data generation, 47, 158
- default, 14, 158
- definition
 - external variable, 42
 - foreign word, 115

- sample, 59
- search pattern, 94
- text unit, 42
- DEUTSCH.BAT, 14
- DEUTSCH.STP, 8, 13, 125
- diacritics, 45, 121, 122, 129
- dictionary, 158, 160
- digit, 158
- diphthong, 14
- disambiguation, 30, 158
- diskette/hard disk full, 138
- DOS-commands, 38
- EBCDIC, 159
- edit, 27
 - line, 38
- EDIT.EXE, 36
- editor, 36
- emendation, 46
- EMM386.EXE, 11, 140
- EMS, 10, 11, 20, 22, 63
- encoding, 45
- ENGLISH.BAT, 14
- ENGLISH.STP, 8, 13, 28, 33, 125
- environment variable, 12
- error, 137
- error messages, 137
- errors with files, 138
- exclusive strings, 83
- Exit, 35
- expanded memory, 20
- explorative method, 125
- extended memory, 20
- external programs, 13, 27, 32, 36
- external variable, 9, 43, 45–48, 52–54, 59, 63, 79, 80, 99, 158
 - definition of, 41
- file, 158
 - 386MAX.SYS, 20
 - ANSI.SYS, 11, 17, 22
 - ASCII, 45
 - AUTOEXEC.BAT, 10–12, 129
 - category label, 103
 - coded text unit, 94, 105
 - CODED-file, 108
 - COMMAND.COM, 22
 - CONFIG.SYS, 10, 11, 17, 20–22, 129, 138, 140
 - DEUTSCH.BAT, 14
 - DEUTSCH.STP, 8, 13, 125
 - DIC-file, 103
 - EDIT.EXE, 36
 - EMM386.EXE, 11, 22, 140
 - ENGLISH.BAT, 14
 - ENGLISH.STP, 8, 13, 28, 33, 125
 - EPSON-LQ.CFG, 18
 - format, 47
 - FRANCAIS.STP, 13, 125
 - FWORTE.DAT, 13, 115
 - HIMEM.SYS, 11, 20, 22, 140
 - HPDESKJ.CFG, 18
 - HPLJ3P.CFG, 18
 - INTEXT.DEF, 8
 - INTEXT.INI, 8, 14, 15, 36
 - INTEXT.MMT, 12, 14, 25, 26
 - INTEXTD.MMT, 14
 - INTEXTE.MMT, 14
 - ISM.DEF, 13, 15, 23, 28, 64, 79, 122, 123, 134
 - ISM.EXE, 32
 - ITXINST.EXE, 10–12, 17, 137
 - JOB-file, 31
 - KEYBOARD.SYS, 129
 - KONTAKT.DIC, 95
 - KONTAKT.ITX, 23
 - KONTAKT.LAB, 103, 120
 - KONTAKT.TXT, 23
 - KONTAKT.WB, 23
 - LAB-file, 103, 108
 - label, 108
 - NEC-P6.CFG, 18
 - NEG-file, 108
 - NEG-POST.DEF, 15
 - NEG-PRE.DEF, 15
 - NLSFUNC, 129
 - output file, 25, 105
 - packing list, 14
 - PRINT.CFG, 15, 19, 137
 - PRINT.DEF, 13, 15, 19, 135
 - project, 35
 - project file, 106
 - QEMM.SYS, 20
 - QUAL.DIC, 120
 - QUAL.TXT, 120
 - RAMDRIVE.SYS, 20
 - rapport file, 108

- raw text, 27
- README, 11
- REFO.SYL, 13, 116
- REFOD.SYL, 13, 116
- REFOE.SYL, 13, 116
- REST-file, 108
- setup, 108
- setup for statistical software, 31
- size, 139
- sort, 139
- SORT.EXE, 13, 36, 139
- STAR2410.CFG, 18
- STARLC10.CFG, 18
- system file, 27
- system of names, 15
- T2I-DIC.EXE, 33, 128
- TAB-file, 108
- tabulation file, 105
- types, 15
- uncoded text unit, 105
- VEC-file, 108
- vector file, 105
- VGAINST.EXE, 17, 137
- VGAMODES.DOC, 17
- VGATMODE.DEF, 15, 17
- wrong format, 138
- file doesn't exist, 138
- file formats, 25
- file names
 - system of, 14
- FILES, 11
- files server, 12
- floating text, 158
- foreign word, 115, 158, 159
- format, 158
 - fixed, 47
 - free, 47
 - input file program ISYS, 55
 - line, 47
 - page, 47
 - paragraph, 47
 - raw data, 158
 - raw text, 47, 52
 - system file, 158
 - TTR file, 147
 - variable, 47, 160
- WORDCOMP-file, long, 147
- WORDCOMP-file, short, 146
- formula
 - ICRC, 106, 107
 - readability, 115
- FRANCAIS.STP, 13, 125
- frequency, 65, 68, 72, 76, 80
 - maximal, 65, 68, 72, 76, 80
 - minimal, 65, 68, 72, 76, 80
- frequency table, 160
- function keys, 22, 37
- FWORTE.DAT, 115
- Giegler, Helmut, 44
- GO words, 33, 63, 89, 116
 - search pattern, 89
- hardware, 17, 18, 20
- help system, 8, 22, 38
- HIMEM.SYS, 11, 20, 140
- homonym, 158
- hyphenation, 46, 47
 - string, 158
- ICRC, 106
- inclusive strings, 83
- index, 79
- infix, 158, 160
- installation, 10, 11
 - network, 12, 13
 - printer, 18
- interactive coding, 94, 106
- interfaces, 39
- INTEXT, 128
 - environment variable, 12
- INTEXT.DEF, 8
- INTEXT.INI, 8, 14, 15, 36
- INTEXT.MMT, 12, 14, 25, 26
- INTEXT/386, 11, 22, 47, 63
- INTEXT/486, 22, 47, 63
- INTEXT/586, 22, 47, 63
- INTEXT/PC, 47
- INTEXTD.MMT, 14
- INTEXTE.MMT, 14
- IRM, 59
- ISM.DEF, 15, 23, 28, 64, 79, 122, 134
- ISM.EXE, 32
- ITXINST.EXE, 10–12, 17, 137
- job generation, 107
- justification, 63, 67, 72, 76, 80, 128

- keyboard driver, 45
- KEYBOARD.SYS, 129
- Klein, Harald, 44
- KONTAKT.DIC, 95
- KONTAKT.ITX, 23
- KONTAKT.LAB, 103
- KONTAKT.TXT, 23
- KONTAKT.WB, 23
- Kramer-Santel, Claudia, 43, 50
- KWIC, 98, 157
- KWOC, 98
- \LaTeX , 134
- length, 65, 68, 72, 76, 80
 - category label, 103
 - maximal, 65, 68, 72, 76, 80
- limitation
 - external variable, 42, 47, 145
 - foreign word, 115
 - length of a concordance line, 99
 - length of category labels, 103
 - RAM, 63
 - search pattern, 95
 - tabulation file, 105
 - text unit, 145
 - vector file, 105
 - word list, 63
- line counter, 13, 14, 36
- line density, 19
- line editor, 38
- line format, 9, 47, 53
- line length, 53, 67, 99
- line spacing, 19
- list of items, 11
- list of uncoded words, 28, 110, 158
- litotes, 94
- machine readability, 45, 46
- margin, 19
- maximal frequency, 65, 68, 72, 76, 80
- maximal length, 65, 68, 72, 76, 80
- memory usage
 - mass storage, 21
 - RAM, 21
- menu, 22, 37
- merge vocabularies, 32, 124
- minimal frequency, 65, 68, 72, 76, 80
- minimal length, 65, 68, 72, 76, 80
- Mittenecker, 31, 118
- MS-DOS, 35
- multiple character, 45, 122
- multiple search patterns, 30
- NEG-POST.DEF, 15
- NEG-PRE.DEF, 15
- negation, 7, 12, 31, 94, 106, 108
- networks, 12, 13
- new words, 83
- NLSFUNC, 129
- NOEMS, 11, 22
- non displayable characters, 134
- numeral, 159
- optical character recognition (OCR), 44, 45
- packing list, 14
- page format, 9, 53
- paper, 137
- paper format, 19, 135
- paragraph format, 9, 53
- parameter field, 94, 95, 103, 128
- parameters, 22, 37
 - program WORDBOOK, 64
 - program WORDCOMP, 84
- personality structure analysis, 31, 59, 118
- pre-editing, 45
- prefix, 95, 159, 160
- PRINT.CFG, 15, 137
- PRINT.DEF, 15, 135
- printer, 11, 137
 - driver, 18
 - EPSON, 18
 - HP DeskJet, 18
 - HP LaserJet IIP, 18
 - NEC, 18
 - Star LC10, 18
 - Star LC24-10, 18
- printer control sequences, 15
- printer driver, 12
- printing attribute, 19
- printing width, 137
- program crash, 140
- program IRM, 13, 17, 33, 135
- program PERSANA, 31
- program REFO, 14, 31, 115
- program SUWACO, 30, 31
- program WOBANA, 28, 29, 63

- program WORDBOOK, 14, 28, 36, 63, 79, 91
- program WORDCOMP, 28, 83, 84
- project, 12
- project file, 35, 106
- project name, 25, 36
- punctuation marks, 47

- QEMM, 20
- qualitative data analysis, 119

- RAM, 35
- RAMDRIVE.SYS, 20
- rapport file
 - coded text unit, 108
 - complete coding control, 108
 - negated text unit, 108
 - program SUWACO, 108
 - uncoded text unit, 108
- raw text, 25, 27, 41, 45–47, 52, 159
- readability analysis, 31, 43, 115, 158
- README, 11
- record, logical, 158
- references, 80
- REFO.SYL, 116
- REFOD.SYL, 116
- REFOE.SYL, 116
- repetition, 31, 118
- restart point, 106
- results, 33
- reverse vocabulary, 6, 29, 33, 128
- reverse word list, 70
- rules
 - fixed format, 53
 - writing, 47

- sample, 9, 35, 54, 55, 59, 63, 64, 67, 72, 76, 80, 91, 99, 101, 108, 117, 119
- SAS, 13, 31, 36, 39, 105, 107, 108
- scanner, 44, 45
- search pattern, 25, 28, 30, 33, 45, 47, 94, 95, 99, 101, 103, 106, 119–121, 128, 135, 157–160
 - ambiguous, 31, 106
 - case folding, 31
 - coding, 31
 - GO words, 89
 - length, 95
 - negated, 31
 - printing, 98
 - searching, 32
 - types, 94
- selection
 - text unit, 59, 64, 67, 72, 76, 80, 99, 101, 108, 117, 119
 - vocabulary, 28, 29, 63, 68, 72, 76, 80
 - words, 65, 68, 72, 76, 80
- sentence marks, 47
- sequence number, 159
- setup, 31, 108
- SIC, 28
- single sheet, 19
- size of output file, 105
- SMARTDRV.SYS, 20
- SOLIS data bank, 54
- sort, 121, 123
- sort criteria, 72, 122
- sort key, 32
- sort modes, 123
- sort order, 12, 13, 23, 28, 63, 79, 121–123, 134
- sort programs, 13
- SORT.EXE, 13, 36
- special characters, 47, 159
- special word, 158, 159
- spelling, 63
- SPSS, 13, 24, 31, 36, 39, 105, 107, 108
- SRM data bank, 54
- starting point, 106
- statistical analysis, 31, 50
- statistical software, 13, 119
- STOP words, 9, 12, 28–30, 33, 63, 68, 72, 76, 79, 80, 159
- string, 158–160
 - exclusive, 83
 - hyphenation, 158
 - inclusive, 83
 - truncate, 160
- structure, 25
- style analysis, 28, 43
- suffix, 95, 159, 160
- supervisor, 25
- system file, 18, 27, 45, 46, 135, 145, 158, 159
- system of file names, 14, 15

- T2I-DIC.EXE, 33, 128
- tab file, 146

- tabulation file, 105
- \TeX , 134
- text mode, 17, 18, 138
 - extended, 138
- text processing, 27, 46
- text unit, 30, 43, 46, 48, 79, 157–160
 - coded, 105
 - selection, 59, 64, 67, 72, 76, 80, 99, 101, 108, 117, 119
 - uncoded, 105
- TEXTPACK, 33, 128
- token, 29, 91, 125, 160
- TRI, 158
- truncate, 160
 - characters, 47
- TSR-programs, 21
- TTR, 29, 63, 64, 79, 83, 91, 125, 160
- TTR dynamics, 91
- TTR file, 147
- type, 29, 91, 110, 125, 160
- type setting, 46
- Type-Token-Ratio, 83

- umlauts, 23, 28, 31, 45, 121, 123
- upper-/lowercase, 45
- uppercase, 94

- vector file, 105, 146
- vertical movement, 19
- vertical text, 160
- VGA text mode, 17
- VGA-card, 11, 17, 137
- VGAINST.EXE, 17, 137
- VGAMODES.DOC, 17
- VGATMODE.DEF, 15, 17
- vocabulary, 8, 18, 28, 32, 33, 63, 89, 103, 128, 135
 - comparision, 83, 135
 - merge, 32
 - reverse, 29, 33, 128
- vocabulary comparision, 13, 86–88

- wild card, 89, 95
- word, 160
- word comparision, 88
- word length, 160
- word list, 13, 23, 28, 29, 47, 59, 63, 64, 67, 69, 71, 121, 125, 145, 158, 160
 - reverse, 67
- word permutation, 6, 8, 12, 29, 30, 32, 59, 67, 75, 78, 89, 145
 - reverse, 67
- word root chain, 30, 94, 95, 120, 159, 160
- word sequence, 6, 8, 9, 12, 13, 29, 30, 59, 67, 71, 74, 89, 94, 103, 145
 - reverse, 67
- WORDCOMP-file, 146, 147

- XMS, 10, 11, 20, 22, 63