Commonalities, differences and limitations of text analysis software: the results of a review

Alexa, Melina; Züll, Cornelia

Veröffentlichungsversion / Published Version
Arbeitspapier / working paper

ZUMA-Arbeitsbericht 99/06

Commonalities, differences and limitations of text analysis Software: The results of a review

Melina Alexa & Cornelia Zuell

Juni 1999
ISSN 1437 - 4110

ZUMA
Quadrat B2,1
Postfach 12 21 55
D-68072 Mannheim

Telefon: (0621) 1246 - 147
Telefax: (0621) 1246 - 100
E-mail: zuell@zuma-mannheim.de
oder alexa@zuma-mannheim.de

INHALT

ABSTRACT

This paper discusses on the one hand the tendencies in functionality and technology of software for text analysis and reflects, on the other hand, on the areas where more development is needed. The basis for this discussion forms a comprehensive review (Alexa & Zuell, in press) of fifteen currently available software for text analysis. In the review each software package, i.e. AQUAD, ATLAS.ti, CoAN, Code-A-Text, DICTION, DIMAP-MCCA, HyperRESEARCH, KEDS, NUD*IST, QED, TATOE, TEXTPACK, TextSmart, WinMAXpro, and WordStat, was presented in a detailed and extensive manner. In this paper we shall only delineate our methodology and criteria for selecting which programs to review and concentrate on discussing the types of support the selected programs offer, the commonalities and differences of their functionality, point to some of their shortcomings and put forward suggestions for future development.

1. Introduction

At present, the user of text analysis software can choose from a fairly rich variety of software supporting text analysis tasks within different disciplinary contexts in considerably different ways. Qualitative-oriented as well as quantitative-oriented and, also, a variety of stylistic, literary and more general text analysis software are currently available which aid the analysis of text data to a greater or lesser extent by providing means for

- organising texts and their coding into projects,
- exploring how frequently and how words are used in context as well as exploring the coding, e.g. how often particular categories have been assigned to a word or text segment, which categories and how often occur together, what links or relations exist between categories or coded text segments, etc.,
- creating and maintaining categories and categorisation schemes
- assigning one or more codes to word strings, words, phrases, sentences, lines, paragraphs or whole texts,
- keeping notes (memos) on text, categories, coded text segments,
- obtaining different 'views' of the text data as well as the coded parts of a text or a group of texts
- exporting the coding for further processing with other software as well as generating reports on the performed analysis and
- supporting team or co-operative work for a text analysis project and merging coded texts.

This article discusses on the one hand the tendencies both in functionality and technology of modern text analysis software and on the other hand it reflects on some of the areas for future development. This can assist users of text analysis software in choosing between available programs, but also in thinking about, testing and enriching their analysis methodology. The basis of this discussion is an already carried out review of fifteen currently available software for text analysis (see Alexa & Zuell, in press). The software considered are: AQUAD, ATLAS.ti, CoAn, Code-A-Text, DICTION, DIMAP-MCCA, HyperRESEARCH, KEDS, NUD*IST, QED, TATOE, TEXTPACK, TextSmart, WinMAXpro, and WordStat. Note that one of the authors of this review is also the developer of TATOE and the other the developer of TEXTPACK. The following programs are typically categorised as 'qualitative': AQUAD, ATLAS.ti, HyperRESEARCH, NUD*IST, QED and WinMAXpro, whereas CoAn, DICTION, DIMAP-MCCA, KEDS, TEXTPACK, TextSmart and WordStat are 'quantitative' ones. Code-A-Text and TATOE support operations which belong to both qualitative or quantitative. One should, however, be careful with this broad categorisation: the selection of the appropriate software should depend on the research question at hand and not necessarily on the above dichotomous categorisation. Besides, when it comes to computational support, the borderline between quantitative and qualitative methodologies may often become 'obscure'; instead, one can detect a number of commonalities which may be placed within a broader text analysis context.

2. Classification of software

Software for text analysis may be categorised according to numerous criteria: For instance, according to the operating system programs run on or the programming language used for their implementation, according to whether they are commercial products for general use or have been developed within academia for specific text analysis research purposes, or according to their suitability for a particular discipline. However, judging from the available literature on the

subject (see Fielding & Lee 1991, Tesch 1990, Tesch 1991, Weitzman & Miles 1995, Evans 1996, Klein 1997a, Roberts 1997a among many) two broad categorisations can be detected: programs are either grouped together according to their functionality focus, e.g. database managers, archiving programs, text searching programs, text retrievers, taggers, code-and-retrieve programs, etc., or according to the type of research orientation they support, that is, quantitative, qualitative, stylistic, literary, etc. analysis.

The categorisation we have used for our review draws a line across the above two, covering software packages which support different types of research within various disciplinary contexts and which are also characterised by *a combination of functionality*: the reviewed software are not solely text retrievers, or text exploratory tools, but rather they support text analysis by a *combination of operations* for coding as well as searching and retrieving text and maintaining schemes and dictionaries for coding.

In particular, operations which are of 'primary' interest during the process of analysing text may be organised into four quick-and-dirty groups:

- *Text import and management*: some of the operations concerned cover importing, opening and saving a text or a number of texts, grouping texts together and managing different text groups, retrieving single texts or text groups and merging coded texts and projects.
- *Exploration*: operations here cover not only text exploration, i.e. information about words or strings of words of a text, but also exploration of the existing coding, i.e. querying and retrieving information about both the categories used and the coded text segments. For text exploration purposes some of the operations involved are generating and/or maintaining a list of all words of a text corpus and counting their frequency of occurrence, creating a word index (text source included), general text counts about word types, tokens, sentence length, etc., searching and retrieving occurrences of particular stems, words and strings of words, proximity searches, Boolean operator-based searches, providing Key-Word-In-Context (KWIC) concordances. Some of the operations for exploring the existing coding concern generating and maintaining a list of all coded segments, Boolean and semantic operator-based searches, KWIC displays for one or more coded segments, perhaps grouped or filtered according to a particular scheme category, and linking and hyper-linking coded segments perhaps with typed relations.
- *Dictionaries, categorisation schemes and coding*: this group comprises operations such as creating a categorisation scheme or a dictionary to be used for coding, and/or importing a scheme or a dictionary, maintaining it, defining relations holding between the categories of a scheme, coding automatically, interactively or manually whole texts and/or specific text segments, assigning notes to texts as well as codes or coded text segments, and merging coding.
- Finally, the group for *export* operations: for example, saving the text only or the coding only or both as well as the categorisation scheme(s) or the dictionary used as in different ways, e.g. raw ASCII/ANSI- or HTML- or Standard Generalized Markup Language (SGML)/eXtensible Markup Language (XML)-encoded files, exporting coding to other software packages, e.g. for statistical analysis, and exporting text to database software.

Naturally, text analysis programs differ according to whether or not they support all the above listed groups of operations: for example, text analysis programs differ according to whether or not they support text exploration, e.g. word searches, word co-occurrence information, etc. More importantly, however, they differ according to *how* they support a certain operation, e.g. all the selected programs support coding, but the way this is realised or the definition of the coding unit differ from program to program.

The list of the above mentioned operations is not exhaustive; there is a large number of operations not catalogued above which, depending on each individual text analysis project, can also be considered primary: for instance, managing and coding multimedia material, support for the transcription process or audio material, etc. Furthermore, clearly, not every single operation of each group and every single group may be required for all projects for each analysis phase, nor has every operation the same importance for all projects.

Still, during the entire length of a text analysis project a combination of operations which belong to the coding group with the operations of at least two of the remaining three groups are required. This claim is based, first, on the gathered user needs as resulted from a survey (Zuell & Alexa 1998) which was carried out among the users of the computer-assisted content analysis program TEXTPACK (Mohler & Zuell 1998), second, on our daily communication with and consulting of researchers designing and carrying out text analysis studies, third, on the main tendencies and practices in computer-assisted text analysis methodology till today as reported in (Alexa 1997), and last, but not least, on the state of the art in text analysis methodology at the moment.

Therefore, we reviewed only those software packages which provide support for a combination of operations of the coding group with the operations of at least two of the remaining three groups. Furthermore, a separate selection parameter concerns the operating system on which a package runs: initially, the operation system of each program was not a selection parameter and MS-DOS, Windows, Mac and Unix running programs were selected. However, we had to take into account that future development does not include DOS-running programs and that the majority of researchers nowadays work with Windows, Unix or Macintosh. Therefore, only programs for these platforms were selected. Note that the review focuses on text material only, and, thus, if a selected package provides multimedia functionality, supporting working with pictures and sound, this feature is solely reported, but not reviewed.

One concern related to this decision was whether the fact that we did not include some older but widely-used programs running on DOS, such as for instance TACT (http://www.epas.utoronto.ca:8080/cch/TACT/tact0.html), we ran the risk of missing important kinds of support or operations which should be considered. In particular, the question which arose was whether there is functionality in TACT or MECA (Carley & Palmquist 1992) that is either not available by, or better than what is available in the Windows or Mac-based programs. For that reason we also considered some of the DOS-programs, e.g. Ethnograph[1] (http://www.QualisResearch.com), INTEXT (Klein, 1997b), MECA, PLCA (Roberts 1989) and TACT at an early phase. This testing has played a role in shaping the group of the 'primary' operations presented above.

[1] At the time of the selection process and the review the Windows version 5.0 of Ethnograph was not available (see also Section 3).

3. Selection process

After having defined the selection parameters discussed above, we compiled a provisional list of all text analysis programs we knew or heard of. Along with the relevant literature (e.g. Weitzman and Miles 1995, Tesch 1990, Klein 1997a, Evans 1996), our additional sources for gathering information have been, on the one hand, Web sites providing links to and information about text and content analysis software and, on the other hand, email communications of relevant user groups (e.g. on qualitative analysis, statistics, humanities computing). This resulted in a collection of diverse information on text analysis software, ranging from product flyers to research papers and user's manuals for some of the programs. The resulting list of software was narrowed down by excluding those text analysis programs which do not support a combination of functionality as explained in the previous section. The outcome was a shorter list of software, which was also more suitable for the purposes of the review.

The material gathered for each selected program provided us with a good idea of the background and motivations for developing some packages, the strengths of each software according to its developer(s) or distributor(s) as well as the types of users they target and the kinds of text analysis projects they wish to support.

We then examined a subset of the selected software in order to 'validate' the list of primary operations we had initially specified. More specifically, in addition to the DOS-based programs mentioned in the previous section, we reviewed the demo versions of ATLAS.ti, NUD*IST, and WordStat, the beta version of DIMAP with the MCCA content analysis module, and the full version of TextSmart. The result of both information processing and software examination, was to slightly complement as well as differentiate where appropriate, the operations we initially specified.

Software selection was not independent from cost considerations: where there was a demo or a free beta version or a test version of a program available, we used this instead of the full version. Demo versions are different in that some are full versions of a program with the restriction that the program may either be used for a limited period of time or its functionality spectrum is exactly the same as the one of the full user version with the exception that it is not possible to either open or import or save one's own texts and analysis or export coding. Software packages for which the licence costs were prohibitive for our 'tight' budget, for example, the user version of CatPac (see http://www. terraresearch.com/catpac.html), were not considered.

Finally, since there was a deadline set for the review only programs were considered which we heard about or whose Windows versions were available by August 1998. However, if patches were made available for a reviewed program after this deadline, we also considered the updates and corrections for the review.
The selection process described above resulted in a total of fifteen text analysis programs, each of which is reviewed in detail in Alexa & Zuell (in press). The review discusses for each program the text import mechanism and the text structure it can handle, the ways each program supports categorisation and coding, its text exploration mechanisms and user interface, its functionality regarding export of both text and coding and their possible interoperability with other programs. In addition, the review provides references to applications of each program for particular studies, to other reviews as well as to program descriptions.

Note that the consequence of reviewing demos or test or beta test versions is that, unfortunately, aspects concerning e.g. the stability of the program when used intensively, or the speed of

processing could not be tested for some programs. We, certainly, consider these to be decisive factors when choosing a program, but, nevertheless we decided to concentrate instead on the examination of how and to what extent the different software packages provide for a combination of functionality for text analysis. It is important to point out, that the demo or test versions reviewed, vary considerably in the kinds of support they offer and stability of the programs.

4. Program capabilities and future development

Given the variety and number of ways by which the reviewed software[2] support text analysis one may suggest that a researcher engaging in text analysis is spoilt for choice: she can import and structure in a variety of ways text data, code text automatically or manually according to a dictionary or a categorisation scheme, search for, retrieve and manipulate text segments as well as coded text, test hypotheses concerning the text material analysed and the categorisation used, export the coding to other software or generate reports on the coding performed. Yet, there are text analysis tasks and methodological approaches, which are not supported enough, or there is functionality needed, which has yet to be developed or integrated in text analysis systems. In the following sections, we reflect on the 'state of the art' in text analysis programs, as this follows from the reviewed programs, and discuss the differences, commonalities, lacking features and possible future development.

It is trite to mention that the programs differ according to the type of analysis approach they have been designed for: what is more interesting to mention, however, is that there are important differences even within the same type of orientation: For example, the coding unit in qualitative-oriented programs may be a line or a word or a sequence of words or the whole text or document; or as another example, dictionary (or dictionaries) which is (are) used for coding texts automatically in quantitative programs may be user-defined or system internal or both. The choice of one or more programs for a specific text analysis project is better served if the user takes into account the *ways* by which analysis operations are supported by the program or programs and the consequences these have for the analysis results, rather than considering solely the general type of analytical approach supported.

4.1. Text import and structure

All programs reviewed read or import ASCII/ANSI-files. Most of the programs dictate their own, often, idiosyncratic text structure delimiters and syntax. In Table 1 each program is listed together with information about the text format of the files it can open or import, the kind of text structure supported, whether texts which have been coded can be imported and whether numeric or demographic variables, i.e. facts, may be imported and used for the analysis.

Table 1: Data Import

Program	Text format	Text structure	Coded text	External facts
AQUAD	ASCII	project, text file, line		

[2] Henceforth, whenever we refer to text analysis programs or software we mean the selection of the fifteen programs which we reviewed.

ATLAS.ti	ASCII	hermeneutic unit, text file, paragraph, sentence, line	coded text as part of project to be merged	
CoAn	ASCII	IDs up to 3 levels		
Code-A-Text	ASCII RTF	text file, speech unit or paragraph		
DICTION	ASCII	text file, each with max. 500 words		
DIMAP-MCCA	ASCII	2 levels		
HyperRESEARCH	ASCII	study, text file with max. 16,000 characters		
KEDS	ASCII	project, text file, paragraph or sentence		
NUD*IST	ASCII	project, text file, header, text unit	coded text as part of project to be merged	demographic data
QED	ASCII	text, data card		
TATOE	ASCII XML-encoded	corpus, text, paragraph	coded text	
TEXTPACK	ASCII	IDs up to 3 levels, sentence		numeric variables
TextSmart	ASCII	question, respondent		numeric variables
WinMaxPro	ASCII	project, text group, text file, line	coded text as part of project to be merged	numeric variables
WordStat	ASCII spread-sheet format	project, respondent, question		numeric variables

Regarding text structure, not all programs delimit the same kinds of structural elements. To give an example: WinMAXpro (Kuckartz 1998) structures texts according to line numbers (numbering is performed automatically during text import) and if hard carriage returns are included in the ASCII-file, these will be interpreted as new lines. NUD*IST (Richards 1998), alternatively, works with text units as opposed to text lines (NUD*IST numbers automatically the text units). Text units in NUD*IST are the smallest units of the document the user may code or retrieve, e.g. a page, a paragraph, a sentence, etc. They are delimited with hard carriage returns (which are typically used for delimiting paragraphs). If it is required that a text unit be other than paragraph, e.g. a sentence, a line or even a word, the text must be prepared so that carriage returns are inserted after the end of each unit. The text files processed by NUD*IST may contain a header and a sub-header: the former involves information about the project, the speakers of an interview, etc. The latter has the purpose of dividing the documents into sections, delimiting each section by means of the '*' character. The user must determine the text structure prior to importing the texts and, hence, pre-processing may be required to structure the files accordingly. CoAn and TEXTPACK accept ASCII-files, which may contain textual 'identifiers', i.e. IDs, denoting text units. Three 'levels' for IDs may be differentiated, whereby every level represents some external feature of the text. The IDs are delimited by special characters, e.g. $ or %, and hard carriage returns can indicate a new text unit with a new ID. Of course, the text files have to be prepared accordingly by the user before using them with the programs.

Two issues are crucial here: first, the mark up for delimiting structural elements of texts and second, which structural elements can be represented and, subsequently, processed by which program. The majority of the reviewed programs neither make use of a 'standard' for the encoding of the different types of texts whose analysis they support nor conform to or exploit the SGML/XML standard for document representation. Instead, each program uses program-specific and idiosyncratic encoding means for marking up structural elements of the documents to be analysed, i.e. text unit, survey question, questionnaire number, header, comment. This means that, firstly, although a considerable amount of time and effort is invested in the preparation (pre-formatting, decisions about which structural elements are important for analysis, etc.) of a text or body of texts for analysis, this text material may hardly be re-used with another program or for another analysis project or secondary analysis purposes. The user is hindered in using other programs or must prepare once more the text(s) for analysis, a task which can be time-consuming and frustrating. We are only aware of the work reported by Kuckartz (1997) on attempts to create a 'standard' format for data exchange between three analysis packages, namely INTEXT (Klein 1997b), WinMAXpro and AQUAD (Huber 1997). As far as we know this has been realised in WinMAXpro but not in the other two packages.

Secondly, no use of standard text encoding may result in losing information, because it may not be possible to represent particular elements of text structure. Of course, this also depends on the quality and flexibility of the standard used. As a way of illustrating the possibility of losing information, consider the MCCA content analysis module (McTavish & Pierro 1990, McTavish et al. 1997) of the dictionary development and maintenance program DIMAP: say that the text to be analysed consists of three presidential debates with each debate involving four speakers. DIMAP-MCCA will not support the definition of a structure unit for the date or number of each debate, and for each one for the speakers (three candidates and a moderator) in a single document. One way round this, is to create separate documents – one for each debate – but if we want to use the analysis apparatus of DIMAP-MCCA for one document consisting of all the debates with the speaker distinction, then this will not be supported. To give another example, irrespective of a particular program: let us assume that the text to be analysed is a transcribed interview and the goal of analysis is at an initial phase the investigation of the questions or comments of the interviewer as well as the answers of the interviewee(s), and at a later phase the examination of only the questions or only the answer texts. If the questions and answers are not represented as separate elements in the text file to be analysed, the program will not know how to switch to either one or both of them in order to code or search for a text segment.

Both AQUAD and WinMAXpro use line numbers for coding and displaying texts. The user of AQUAD must make sure that each text line contains no more than 60 characters, whereas WinMAXpro enables the user to set herself what the length of a line should be. In both cases the user can code line-by-line. In WinMAXpro, each code is displayed together with information about the number of lines coded it has been assigned to. The assumption which appears to be made here concerning line-based coding is that text lines can be significant or carry some semantic weight with regards to coding. Although this might be true for particular genres, e.g. poems, it is doubtful for transcriptions of such texts, as interviews, discussions or debates, answers to open-ended questions, articles, etc.

Some of the programs impose restrictions on the size of a text unit or the automatic structuring in text units or text lines. Text to be processed by DICTION (Hart 1985) must not be longer than 500 words. If the researcher wishes to analyse longer texts, DICTION splits them in text passages of 500 words. The reason behind this relates possibly to the calculation of standardised

scores of DICTION. Nevertheless, the statistical basis for this restriction is not clear and as far as its 'semantic' basis is concerned we cannot see any.

In other words, the amount and detail of structure allowed by a software, that is, what structural elements of text are represented and recognised, is a deciding factor for the types of operations which can be performed: only explicitly represented elements can be searched or operated on. A poor or inadequate representation of structure means that one should expect limitations in the number of operations.

4.2. Categorisation and coding

The meaning of a code or a category varies considerably: categories can mean interpretative codes and may or may not be organised in a scheme used for coding text. A different type of category for coding is what Code-A-Text calls numeric scales based on frequency of occurrence information. These codes are automatically generated by the program. Furthermore, codes can also denote text or case variables, e.g. the age of the interviewee or the date an interview was taken or the name of the newspaper whose articles are analysed. Table 2 categorises each program according to what mode(s) of coding it supports, the kind of dictionary or coding scheme used for coding and whether multiple, i.e. overlapping or nested coding is supported.

Table 2: Categorisation and Coding

Program	Mode of coding	Dictionary	Scheme	Multiple coding
AQUAD	manual		hierarchical	yes
ATLAS.ti	manual, automatic in connection with search		hierarchical, or network-oriented	yes
CoAn	automatic, interactive	word based		yes
Code-A-Text	manual, automatic	word lists, nominal or ordinal scales	flat	no
DICTION	automatic	hard-coded, word-based		no
DIMAP-MCCA	automatic	hard-coded, word-based		no
HyperRESEARCH	manual, restricted automated		flat	yes
KEDS	automatic, interactive	domain specific, word- and rule-based		no
NUD*IST	manual, automatic in combination with search		hierarchical or free 'nodes'	yes
QED	manual, automatic in combination with search		flat	yes
TATOE	manual and automatic in combination with search		hierarchical or flat	yes
TEXTPACK	automatic	word-based		yes
TextSmart	automatic, interactive	automatically generated or user-defined categories, word-based		yes
WinMaxPro	manual, automatic in combination with search		hierarchical	yes
WordStat	automatic, interactive	word-based		no

Categories in NUD*IST are called nodes and they cover both interpretative categories and variables. Both types of categories are organised in the Index Tree, which is a hierarchy of the categories created in NUD*IST for coding. There are no overarching, more general categories or, alternatively, families or schemes to organise categories separately. A possible disadvantage of organising categories in that way is that the Index Tree may grow tremendously, making in that way the keeping of an overview over the interpretation and the handling a large amount of information as one big block difficult. Because the Index Tree comprises categories for both 'overall text' coding and 'interpretative' coding, it may become fairly 'lumpish'.

The identity stamp of a category differs depending on both program and the text analysis approach supported: categories are given a name and for a number of quantitative-oriented programs categories are given a numerical value. This is used at the later phase of exporting the

coding for further statistical processing. The categories in NUD*IST have each a name and an 'address', i.e. the node's numerical value in the coding hierarchy. Most of the qualitative-oriented programs support the building of category hierarchies or networks of categories, the quantitative-oriented ones, typically, use for coding a dictionary with a collection of word lists: the words of each list build a semantic 'category' or a concept and each word is given the same numerical value within the list. Dictionary word lists are not explicitly related to each other.

Whereas categorisation schemes are often constructed by the analyst(s) for a specific analysis project the applicability scope of dictionaries used for automatic coding may be broader. However, domain specific dictionaries may form the core part of the program for automatically coding and categorising a body of texts, e.g. DICTION and MCCA, in which case their ideological and domain dependence cannot be avoided.

The categories may be a simple collection of codes organised with a project for coding a group of texts or they may be organised in a hierarchy or as a number of hierarchies, e.g. WinMAXpro or ATLAS.ti (Muhr 1996, 1997). Only one of the reviewed packages supports the definition of relations holding between categories which are not hierarchical: the user of ATLAS.ti may use a variety of up to seven types of semantic relations, called links, such as, cause_of, part_of, etc., to link two codes or quotations. Moreover, the program supports the user in defining new relations if the ones offered do not cover the user's needs with the restriction that user-defined relations are possible only for code-code or quotation-quotation links. The user is guided through different options in order to define a new relation so that the system can make appropriate use of it in the subsequent processing.

Dictionaries can be lists of words assigned to a number of categories or lists of words organised in categories, which, in turn, are organised in super-categories. The dictionary entries have no relation to each other, they are just elements of a set.

A further way by which dictionaries may be differentiated concerns the types of entries they contain. As an example, consider the dictionaries and the respective rules used by the KEDS (Schrodt 1998) text analysis program for the automatic as well as semi-automatic analysis of event data as these are found in English news lead sentences. KEDS uses three main types of information, organised in separate files, for verbs, actors (mainly nouns and proper nouns referring to political actors) and phrases, for distinguishing between the different meanings of a verb and for providing the rules for locating the source(s) and target(s) of an event within a sentence. KEDS focuses not only on the identification of types of concepts in a text, but rather on identifying and explicitly encoding the connections between the concepts of a news lead sentence.

Contextual parameters or overarching categories may be used in combination with specific dictionaries for guiding the analysis and scoring process in some quantitative programs. The MCCA dictionary, for instance, uses the contextual parameters traditional, practical, emotional, analytic to calculate context scores (called *C-scores),* which capture the structuring of ideas and may determine different social contexts, whereas DICTION makes use of five super-categories, i.e. activity, certainty, realism, optimism, commonality, as 'the main' dimensions or features for describing the language of political speeches (Hart 1985: 111).

Programs which code texts automatically on the basis of a dictionary differ with regards to whether the dictionaries are system-internal and have been prepared by the developer(s) of the

particular program, e.g. DICTION and MCCA, or whether they are external in the sense that the user creates and provides the dictionaries used. Beside this distinction, such programs may enable the user to modify the supplied dictionaries (see for example KEDS or WordStat) or only allow the addition of user dictionaries to the ones mainly used by the system, as DICTION does.

For a more targeted analysis or for processing and storage efficiency reasons quantitative programs may use stop word lists, i.e. lists of words which often occur very frequently and whose content is not assumed to be important for the analysis purposes and, therefore, should not be considered for analysis: for example, the excluded terms list of TextSmart or the exclusion list of WordStat. Stop word lists may be provided either with the program (e.g. WordStat or TextSmart) and may be modified on- or off-line by the user, or, alternatively, the user specifies a file name for a file containing the words which the system should not consider, as is the case with Code-A-Text or TEXTPACK. Note that a stop word list may be used for text exploration purposes rather than coding: the WordCrunch function operation in ATLAS.ti generates a word list containing type/token ratio information and the frequency of occurrence for each word of a primary document. If the user wishes to specify a stop word list as an external file which may contain both full words and words with wildcards (e.g. *ask**), ATLAS.ti will not consider these words when counting word frequencies and will not include them in the generated word list.

A stop word list created specifically for a given research purpose is often not appropriate for another analysis purpose, since the high frequency of occurrence of a number of words alone cannot determine whether each word should be included in the stop list and, therefore, not be considered for coding purposes. For example, the words 'I' or 'me' may appear frequently in the text corpus analysed and may be irrelevant for a particular analysis project and, thus, included in the stop list. However, if a text analysis project examines 'self reference', these words can be significant for the analysis. The development of such stop lists which suit the aims of a specific analysis project is not a trivial matter. Software should aid the user in creating such lists (if these are required for analysis), for example, by providing word frequency lists for a specific text, word distribution and co-occurrence information, etc.

If the merging of two or more dictionaries is required, then this operation is not supported by the majority of the software supporting dictionary-based automatic coding. A promising way for avoiding this shortcoming has been put forward with the connection of DIMAP with MCCA. DIMAP is a software for dictionary development and MCCA is a content analysis component of DIMAP. This may profit dictionary-based content analysis. However, the strengths of this approach are still to be demonstrated.

The selection of the programs we reviewed does not include a program for the type of linguistic text analysis described by Roberts (1997b) or Franzosi (1997), or the map analysis described by Carley (1993)[3]: For linguistic/semantic analysis the identification of key concepts or categories, such as actor or subject of an event, is not enough, but rather the linguistic/semantic connections between the concepts of a sentence or clause must be explicitly encoded, on the basis of a semantic grammar. KEDS is the only software program we have reviewed which can be used for the extraction of a set of related concepts, namely targets and sources of an event, according to a set of rules. In map analysis the main aim is, given a set of texts and a set of concepts, to determine for each text whether and which of these concepts occur, as well as determine the

[3] Note that there are no Windows versions for PLCA, PC-ACE (Franzosi 1990), and MECA which support respectively these types of analysis.

cognitive relations between the occurring concepts. As mentioned above, only ATLAS.ti allows the user to define a number of relations between concepts or categories and display them as a network. Still, ATLAS.ti is better for building one's own concept network and theory than studying the specifics of someone else's conceptual network. Besides map analysis as delineated by Carley (1993) and Carley & Palmquist (1992) takes a quantitative approach, aggregating concepts and relationships, a feature which is not supported by ATLAS.ti.

The user of ATLAS.ti may organise categories in 'code families', whereby one category may belong to several code families. This way the user may organise codes and their respective coded segments in meaningful groups. At the same time she has the possibility to combine codes from different families for searches or coding, which is helpful both for text exploration and further coding. In comparison, categories in TATOE (Alexa & Rostek 1996, Rostek & Alexa 1998) are organised in schemes and each category belongs to only one scheme. There are no relations holding between the categorisation schemes used. The definition and usage of multiple schemes may be seen as a way of creating different layers of interpretation on the same text corpus. For manual coding only one scheme can be used at a time. However, the user of TATOE may define search patterns, i.e. rules for sequences or alterations of different categories of different schemes including also strings, and then, if required, code in one go according to a given scheme the retrieved instances (see Rostek & Alexa, 1998). This way, TATOE supports multi-level coding (see also Alexa & Schmidt, in press): Because different types of coding, e.g. interpretative coding and part of speech categorisation, provide additional means for searching the text data, validating coding, and working towards theory testing and building.

4.3. Export and reusability

Most quantitative-oriented programs will export the coded data to a statistics package for the purposes of further analysis. From the qualitative programs ATLAS.ti, WinMAXpro and Code-A-Text (although not clearly qualitative) provide this functionality. In Table 3 each reviewed program is categorised according to whether and if so in what format one may export or save either the text data or the dictionary or categorisation scheme or the coding alone, or whether the text data together with the existing coding may be exported or saved.

Table 3: Data save and export

Program	Text	Dictionary/ Categorization Scheme	Coding	Text and Coding
AQUAD				
ATLAS.ti	ASCII	ASCII	ASCII, SPSS	HTML (whole project)
CoAn		ASCII	ASCII, SPSS	HTML
Code-A-Text			ASCII, SPSS	
DICTION			SPSS	
DIMAP-MCCA			ASCII, SPSS, KYST	
HyperRESEARCH		ASCII	ASCII, SPSS	ASCII (but not side by side)
KEDS		ASCII	ASCII	
NUD*IST	ASCII	Decision Explorer, Inspiration	ASCII	ASCII (but not side by side)
QED				ASCII
TATOE	ASCII, XML	ASCII	ASCII, SPSS	ASCII, XML
TEXTPACK	ASCII	ASCII	SPSS, SAS, ASCII	ASCII
TextSmart			ASCII, SPSS	
WinMAXpro		ASCII	dBase	ASCII
WordStat	ASCII	ASCII	ASCII dBase	

Most qualitative programs maintain a system file encompassing some or all objects of the analysis project, such as texts, coded segments, memos, categories. A number of different system files of a given text analysis package may be merged, supporting in that way team work. If for example, different individuals are analysing different texts for the same project using the same categorisation scheme, they can merge their analysis. This, of course, does not exclude the possibility that a single user can use this functionality in order to manage large projects. It is important to note, however, that this facility is program dependent. Teamwork and merging of projects and their respective files is supported by ATLAS.ti, WinMAXpro and NUD*IST as long the users use the same package. This means that the purpose of this facility is not general exchange, re-usability and merging of projects, but rather local within a particular package.

One of the programs reviewed, namely AQUAD, does not support export of coding at all: The user can neither save separately or export the categorisation scheme nor export the coded text with the framework for purposes of further processing, secondary analysis, etc. We have already mentioned in section 4.1 that Kuckartz (1997) reports on attempts to create a ‘standard’ format for data exchange between INTEXT, WinMAXpro and AQUAD. This would mean that the coding performed in AQUAD would be saved in a way that would make the coded data exchangeable. However, as far as we know this has not yet been realised in AQUAD.

Unfortunately, and similar to our discussion on text import, no standard is agreed upon for either exporting text only or text and coded segments or the categorisation scheme used or all three types of information. The dissemination of raw as well as coded textual data according to a text standard is hardly supported by the software reviewed. CoAn and ATLAS.ti generate HTML-

encoded reports with the coded text and the categorisation scheme, ATLAS.ti will also export memos to XML and TATOE exports text and encoding as XML-encoded data.

Exporting the text analysis data in a standard format would facilitate re-usability of text data and coding. It could enable other researchers with similar analysis purposes and using different packages to use the same text material or compare coding. As a consequence of no agreed standard, the exploitation of already existing data or the validation of analysis or even secondary analysis projects, which do not employ the same program, are hindered.

A further point related to the export possibilities of the reviewed programs concerns the re-usability of dictionaries and - if available – their rules as well as categorisation schemes. For example, the KEDS dictionaries, that is the separate files containing actors and verb patterns and the rules for these patterns, are external to the program and may be edited during analysis. This means that, if required, dictionaries may be re-used for other analysis projects, although some pre-editing will probably be necessary as the structure recognised by each program is not the same. TextSmart, on the other hand, saves the exclusion and alias word lists as ASCII-files, but only those aliases created by the user and not those which were automatically created by means of lemmatisation (see section 4.4) are included. The categories themselves in TextSmart cannot be exported and re-used in other projects, because the categories are directly based on the inclusion word list which is generated for each text on the basis of the words in a text.

Finally, NUD*IST is the only reviewed program which provides a utility program for translating NUD*IST projects between Macintosh and PC platforms: with the QSR Merge utility program NUD*IST supports teamwork and provides multi-platform functionality.

4.4. Linguistic information

Two programs, namely KEDS and DIMAP, make use of linguistic information and categorisation. Although not a full syntactic parser for English, the KEDS program for the analysis of news lead sentences uses pattern recognition to parse particular parts of each sentence, such as verbs, objects of a verb phrase and proper nouns. Sentences to be analysed are both domain and text-type restricted and the KEDS parsing mechanism makes use of these restrictions. Rather than identifying only specific types of concepts in a text, KEDS also identifies as well as explicitly codes the connections between the concepts of a news lead sentence. It uses three main types of information, organised in separate files, for verbs, actors (proper nouns referring to political actors) and phrases, for distinguishing between the different meanings of a verb and for providing the rules for locating the sources and targets of an event within a sentence.

As a dictionary maintenance program, DIMAP supports the definition of different senses of words. It keeps homographs separately and supports the definition of semantic restrictions encoded in word senses. The user may specify hypernym and hyponym links, attribute-value feature structures, and semantic links to create a semantic network. Unfortunately, the advantages of combining a dictionary maintenance program with a dictionary-based content analysis program such as MCCA cannot yet be demonstrated in practice, as “the real integration (where the resources of DIMAP can be used to extend the dictionary used in MCCA) has not yet

occurred. There are several theoretical issues involved (primarily the question of how to perform automatic reweighting)."[4]

Other kinds of linguistic information utilised by some quantitative-oriented programs include lemmatisation or stemming, part of speech categorisation and word disambiguation techniques. Stemming refers to the matching of the beginning of different forms of (possibly) the same word, e.g. 'say', 'saying', says', to a string, called stem. Lemmatisation, in contrast, refers to the matching of all different forms of a word regardless of whether its root is the same, e.g. 'say' as well as 'said' have all the same lemma.

TextSmart is the only reviewed program, which features an integrated automatic lemmatisation for English, but only for the most frequent words[5]. When the text file is first imported and each time the texts are edited, TextSmart lemmatises automatically the text data and creates an alias, that is a word group, with the different word forms identified by the lemma. Stemming is used for the analysis, but not performed by other programs, e.g. CoAn, KEDS, TEXTPACK, where stems are dictionary entries, indicated by a particular character. This means that the user has already specified which words are to be treated as stems. WordStat uses its Substitution Dictionary which, in principle, contains the scheme categories, but may also used for defining lemmas, for example, the word forms 'art', 'arts' and 'artistic' are related to the category 'art'.

A small number of programs uses part of speech information: For example, as noted above, the KEDS program organises the data files it uses for coding according to verbs and proper nouns or nouns. When information about the part of speech category of the words of a text is used by an analysis program, then such linguistic information serves as a means for further categorisation. Note that usually such linguistic information is used only for particular part of speech categories and not for all the words of a text. In other words, morphological or part of speech tagging itself is not part of the program's functionality. For those programs which incorporate part of speech information, so far we have been able to check from the available literature on them, there are no pointers to or information about whether part of speech categorisation has been performed manually or automatically during the dictionary construction. Another way of making use of morphological or part of speech information is as this is supported in TATOE: Texts which have been previously automatically tagged by means of a tagging program may be imported in TATOE as coded texts, and the part of speech categories are stored as a separate categorisation scheme.

DIMAP-MCCA and DICTION make use of disambiguation procedures based on frequencies of occurrence. In particular, MCCA uses context information and the expected frequency of occurrence within a given context for a word; in contrast, the use of frequency information in DICTION is based on counts of homographs differentially weighted within the various dictionaries by applying (a priori) statistical norms for their use. Unfortunately, we have not found references on tests made to measure the accuracy of disambiguation procedures of both of these programs. Systematic evaluation of the content analysis disambiguation or comparisons of the disambiguation techniques in content analysis with more linguistically oriented disambiguation ones are both areas where more future work is necessary to be invested.

---

[4] Ken Litkowski, June 1998, personal communication.

[5] It is worthwhile to note that the TextSmart User's Guide contains no information either about which these words are or the method used for determining high frequency of occurrence.

The recognition and – when necessary – the respective coding of negation is very weakly supported by programs for the automatic coding of texts based on a dictionary. For instance, a few of the DIMAP-MCCA dictionary categories are heavily slanted toward negative words. INTEXT, which due to the fact that it is a DOS-based program was not selected for the review, uses lists of words, stems or word suffixes, which indicate negation expressions, e.g. 'not', 'no', 'never', 'un-', which the user constructs to deal with the recognition of negation.

The majority of the programs go hardly beyond the text, the meta-textual information included in headers and the factual, case- or project-related information. Lexical, syntactic or semantic information, e.g general dictionaries, semantic databases, thesauri or ontologies, or other additional text sources (text corpora) is rarely used or coupled with the analysis. The exception to this is the work reported by McTavish et al. (1997) on using the WordNet (Miller et al. 1993) semantic database for English for examining the MCCA categories: this has shown that the MCCA dictionary categories are consistent with the WordNet synsets (sets of synonyms connected with each other with a number of semantic relations, e.g. part-of, hyponymy, antonymy, etc.). In particular, the usage of the WordNet data is shown to be advantageous in two main ways:

- it allows the analyst to move from subjective conceptual or contextual groups to more general semantic groups which reflect fine-grained meanings inherent in particular words and
- it enables the further refinement of the idea categories into semantic components: it is possible to extend the amount of words that might be associated to an idea category, mainly by identifying and including the narrower terms, i.e. hyponyms.

Both of the above ways may result in explicit and transparent analysis. In general, enriching text analysis with such 'external' linguistic information may have advantages for the exploration of texts, the construction of categorisation schemes and dictionaries, the testing of their validation, and may contribute in more fine-grained characterisation of concepts and themes and also in analysis based on more rigorously defined principles.

One way for expanding the applicability of the programs employing language-dependent linguistic information and techniques for analysing texts in other languages, may be to opt for more modularity: the language-dependent processes of a program are separate modules and the program offers (documented) interfaces to these. This would also accommodate the possibility that interested researchers may (develop and) integrate their own routines (e.g. as DLLs) for the specific language and grammar-dependent parts of the program.

4.5. Exploration of text and coding

Exploration refers to the mechanisms which the programs provide for searching, retrieving, linking and counting frequencies of words or phrases, single codes or combinations of codes according to stated conditions, texts as well as searching and retrieving in memos. Programs differ according to whether their focus is on supporting exploration of coding or exploration of text data as opposed to coding only: on the one hand, qualitative-oriented programs appear to concentrate mainly on exploring coding and some of them support complex and advanced searches, based on stated conditions, for co-occurrences or juxtapositions of codes. On the other hand, quantitative programs tend to support general or Boolean text searches for words or word strings, and display the retrieved occurrences in KWIC lists, where the user can see how the

searched words or strings are used in context, provide frequency of occurrence information, and support code exploration fairly rudimentarily.

In contrast to qualitative programs, the majority of which provides a variety of fairly advanced techniques for searching and exploring coding, e.g. the QueryTool of ATLAS.ti or the Index Search operators of NUD*IST or the Logic Machine facility of WinMAXpro or the possibility to build search expressions for hypothesis testing in HyperRESEARCH, quantitative-oriented programs are weak concerning searching and exploring the coding performed. The assumption which appears to be made is that if the coding results can be exported to a statistics package then coding can be further explored there. However, better possibilities to explore and check coding are important for both building and validating categorisation schemes and dictionaries, and would improve the functionality of these programs.

A feature missing from the majority of the programs we examined, with the exception of Code-A-Text and TATOE, and which is restricted to searching with the OR operator in ATLAS.ti, is the possibility to construct searches which enable to specify sequences or alternations of codes *and* strings (see Rostek & Alexa (1998) on how this is realised in TATOE).

A significant kind of support that a text analysis program should provide for the building of categorisation schemes or the construction of dictionaries, as well as for checking their relevance and validity for a particular text analysis, concerns occurrence and co-occurrence information as far as the text alone and the coding it entails is concerned. In particular, it is often helpful to be able to count some or all words or word stems which occur either to the left or to the right of a particular word or a string of words, or even on either side, within the same 'text unit'; rather than the text analysis program dictating the size of the text unit considered for calculating co-occurrences, it should be possible that the user specifies every time whether the unit is a sentence, a paragraph or the whole text. Note that the possibility to realise this is dependent on the text representation used.

The result of a search for a word, phrase, code or combination of code pattern, can also be displayed as a KWIC list surrounded by the left and right context of the word or group of words searched for. This is useful for getting a quick impression of how the matches are used in context of particular types of texts. WinMAXpro displays the result of a text search as a KWIC display, with the restriction that the user cannot view this on-line, but rather she can save the retrieved instances as an ASCII file. In contrast to other qualitative-oriented packages the WordCrunch option of ATLAS.ti generates a word list of all words occurring in a single (selected) text of a project, together with each word's frequency of occurrence and the type token ratio for this document. Similar to the KWIC option of WinMAXpro, the WordCrunch output can be saved in a file. Alternatively it may be sent directly to a printer or be viewed in an editor. In contrast to TATOE, the KWIC displays of WinMAXpro or ATLAS.ti are not interactive: the user can view, print or edit the KWIC display with the retrieved matches. Since there is no linking between the system and such a display, she cannot select further a word or a string of words from that list and generate a further KWIC list as is possible in TATOE.

An important feature for exploration purposes concerns what one can call 'hypertext functionality' of the reviewed programs. In principle, we may view the code-and-retrieve functions of programs, such as WinMAXpro, ATLAS.ti, QED or TATOE as hypertext functionality: links may be exist among

- categories within a single text or a number of texts

- categories themselves, since relations, e.g. hierarchy, cause_of, etc., may be defined to hold among them,
- texts, codes, coded segments to (pop-up) comments, notes or memos
- categories to families, groups, parallel schemes, e.g. ATLAS.ti, TATOE.

Added to the above are graphs of hypertext-like networks, e.g. ATLAS.ti, or word indexes or lists for a text or a text corpus, where selection of a word displays it as it occurs in the text, e.g. TATOE, or means for investigating the relationships between words, categories and responses, e.g. TextSmart's 'brushing' facility which when switched on, each time the user clicks on a category or a text or an term the selection is broadcasted across all the other windows. All these different kinds of linking mechanisms are means for navigating from one type of link to the next and assist in exploring the text and its coding.

4.6. User Interface

More than half of the programs we examined may be described as 'second generation' text analysis software with a Windows 'look', which make little use of modern graphical user interface (GUI) technology and, more crucially, the 'interactivity' and linking between the displays (different windows) of text, categories and coded instances or search results is either only very basic or not supported. Often, the user runs batch routines organised in menus. ATLAS.ti, QED, TextSmart, TATOE or WinMAXpro are exceptions to this.

The majority of the reviewed programs move often from performing an operation directly to generating a file, instead of including an intermediate level of presentation and browsing as a means of working with the results. An output consisting of a hardcopy-like grouping of all hits after a search operation, with either line numbers indicating the positions of each hit, or each hit listed in a KWIC display, with the left and right context of a word being hard-coded, is perhaps appropriate for saving it as a text file, but it is not necessarily fit for working with the information it includes. Future development will need to provide more flexible ways of browsing and manipulating text displays and retrieved occurrences: further on-screen selections, adjustment of the amount of context displayed, varied displays with coding switched on or off, etc. Text displays and, in general displays of the output of a search should serve the purpose of presenting different views on the data as well as keeping the user 'close to the text data'. The generation of 'lump-it-all-in-one' reports is one of the possible ways of saving analysis results, but not *the* best way for displaying the results on screen and working with text. During text analysis the user is considerably supported if she can see her text and its codes clearly and easily on screen, have direct access to or point-and-click on text segments and perform varying operations. Programs such as AQUAD, NUD*IST or DICTION keep the user at a distance from her text during analysis.

An important aid to analysis is to be able to have different views of the data side-by-side and still have the possibility to perform further operations on them, for example for juxtaposing different kinds of information. Few programs, e.g. ATLAS.ti, TATOE, TextSmart or WinMAXpro, provide for the integration of different display means: usage of margins or highlighting, hypertext links, icons, visualisations of graphs and networks, etc. Future text analysis software need to invest more effort on GUI design using various display means and modalities in a meaningful and user-friendly way.

4.7. Integration of modes and methods

As stated by Tesch (1990: 28-29) on her review of qualitative software "[...] any given concrete study might well incorporate aspects from more than one research type. [...] the specific research

purpose and nature of the data can differ from study to study even within one approach, necessitating unique data management manoeuvres." Weitzman and Miles (1995: 334), spelling out some "cut-across needs and hopes" for qualitative analysis software, affirm that "there is a growing agreement in the qualitative research community that linking qualitative and quantitative data in study designs and analyses can enhance validity, develop richer analyses, and lead to deeper theoretical insight (...)." Furthermore, Stone (1997: 50) argues that "technology no longer constrains researchers to one or another mode of analysis. Instead it is feasible to consider a platform of desktop computer text analysis capabilities, that allows the researchers to draw upon sequences of procedures that best suit their proclivities, to follow up on leads as they uncover textual patterns, and to check the validity of their discoveries."

The incorporation of aspects of different research types, as well as the linking of qualitative and quantitative data and the integration of different modes of analysis are not only uncontroversial, but rather valid methodological decisions for particular text analysis projects. However, few of the reviewed software packages facilitate integration or linking of text analysis methods and modes of coding.

Moreover, few programs support a combination of coding modes (see Table 3), i.e. automatic, interactive or manual. Automatic coding means that the codes are assigned to texts by the computer mainly on basis of a word list or dictionary or a number of them. In addition to a dictionary, rules or a grammar may be used, as in KEDS. Quantitative-oriented programs support mainly automatic coding and only some of the programs of this type support interactive coding: KEDS and CoAn control the automatic coding process and allow the user to confirm coding if necessary, whereas the user of TextSmart, or WordStat or Code-A-Text can select a word or a text segment and assign codes manually independent of the automatic coding process.

Different types of 'automatic coding' are supported by ATLAS.ti, NUD*IST, QED and WinMAXpro which are possible only after a search-and-retrieve operation. In particular, with the Auto Coding facility of ATLAS.ti the user can code either a string matching exactly the search string, or a word or a line or a sentence or a paragraph or the whole text which has been retrieved for a single primary document, i.e. the one the user displays at the time, or for a family of or all primary documents of a hermeneutic unit according a selected code combining thus text search and coding. WinMAXpro offers an option to code all lines where a search string occurs in one go. The user of QED may assign codes to a subset of texts, as opposed to text lines, which have been retrieved after a search. NUD*IST keeps a log of searches and retrieved results, and the user may create a node, a category for the particular search, which means that all the text units retrieved are marked according to the new node.

4.8. Generality of use

It appears that all programs have been developed and designed with concrete application projects in mind. As it may be expected, there is no 'ideal' package general enough for all kinds of text analysis. However, there are degrees of generality of use: The functionality of the majority of the quantitative programs reviewed is often restricted to the analysis of one particular text type, for example, TextSmart is designed only for the analysis of survey texts, i.e. answers to open-ended questions, or KEDS efficiency and functionality lies in its exploitation of a single text type, namely that of event reporting in news leads. In general, software packages differ according to whether they are characterised by a survey analysis orientation or not: survey-oriented ones are designed to handle relatively short texts, code them mainly automatically and export the coding to a statistics package. One can, also, distinguish between qualitative-oriented programs whose

strength lies more in the analysis for dialogue texts, e.g. Code-A-Text, and those programs which support the analysis of a variety of text types, e.g. ATLAS.ti.

The generality of usage of the software is not independent of the type of categorisation a program assumes or supports. Programs whose categorisations are system-internal and whose processing involves calculating scores for the pre-defined, theory- and domain-bound categories have a narrower applicability spectrum in comparison to those which depend on the user to provide the categorisation scheme or dictionary. DIMAP-MCCA belongs to the former. The MCCA dictionary is used in order to produce on the one hand scores for measuring social context according to four parameters, namely Traditional, Practical, Emotional and Analytic, and on the other hand individual emphasis scores for 116 idea/categories. These categories were developed judgementally and are considered important for the analysis of naturally occurring verbal material.

A further example of a software with a 'hard-wired' dictionary is the DICTION analysis program. DICTION is designed specifically for political discourse analysis. It measures the style of political texts according to five general categories, namely Activity, Certainty, Realism, Optimism, and Commonality, based on individual scores for twenty-five sub-categories, assumed as important for the interpretation of this kind of texts.

The text to be analysed comes, of course, in a particular language using the respective alphabet characters of that language. These are both decisive factors for the degree of applicability of a text analysis program. Software which does not support the display and the processing, e.g. alphabetic sorting, stemming, etc., of texts in non-Latin based languages, using other alphabets and characters, can, of course, not be used for projects which either concentrate wholly on text material in one other than Latin based language, e.g. Hebrew, modern or ancient Greek, Japanese, etc., or use texts from different languages, for example Russian and American English reports of the same event, Swedish, Norwegian and German texts which are translations of each other, etc.

5. Summary and concluding remarks

This paper has discussed the capabilities of a selection of text analysis software which support analysis by a combination of operations for manual or automatic coding *as well as* search-and-retrieve functionality and the maintenance for both categorisation schemes and dictionaries. Furthermore, this paper has identified and discussed specific areas where more development is needed. This discussion can aid the choice between available programs and, also, assist text analysts in thinking about, testing and enriching their analysis methodology.

One of the findings of the review we carried out is that very often a single text analysis software package alone supports only *some* of the operations required for a particular text analysis project: For example, analysis with KEDS aims at recognising and coding automatically text units referring to particular events. It does not support more explorative operations such as text searching, linking, etc. Alternatively, programs may not support certain operations efficiently, elegantly or user-friendly enough. This lack of support would not necessarily be problematic if it were possible to use two or more different software packages for a single text analysis project in a seamless and user-friendly way. For certain projects it makes sense to work in an 'integrating' manner, that is, to apply different types of approaches and techniques for instance, testing purposes or for juxtaposing and comparing the analysis results obtained.

However, the researcher wishing to integrate different modes of coding or different methods is, unfortunately, not spoilt for choice (yet?). The merits of different types of methodology and approaches are

- either not integrated in a single modern software to assist the researcher to concentrate on the interpretation task, the building of theories and testing of their validity,
- or cannot be coupled by using two or more text analysis programs for the same body of texts, re-using, and perhaps, extending, existing coding or categorisation schemes or experimenting with different text exploration techniques
- or, finally, can be linked but only if the user has invested or is willing to invest a considerable amount of work in pre- and post-processing.

As far as the future development of text analysis software is concerned, we believe that aiming for a single 'ideal' software for all kinds of text analysis and all projects is unattainable: It is doubtful that we can take into consideration or plan in advance all types of texts and all methods of text analysis possible to be performed and thus produce a complete list of all the tasks a text analysis program could or should support as well as specify how the program should support them. More attainable is to aim for an 'open' and extensible model for text analysis software as a means not only for dealing with text data which were not taken into consideration during the design phase of the program's development, but also for extending the functionality of a program by either adding 'modules' for specific analysis tasks, e.g. disambiguation modules, tagging, etc., or communicating with other programs or using available resources for analyses which were not anticipated at the development phase.

This paper does not propose that text analysis operations should be uniform. We do believe, however, that agreement on or development of standards for text encoding for text import and export purposes, as well as some agreement on a kind of standard terminology for basic concepts in text analysis software, would improve interoperability, contribute to both an increase of usage of text analysis programs and a better understanding among researchers using text analysis programs. Finally, we hope that this discussion contributes to the determination of the more or less 'core' or 'basic' functionality that a 'good' text analysis program should offer.

References

Alexa, M. (1997). Computer-assisted text analysis methodology in the social sciences. ZUMA Arbeitsbericht, Nr. 97/07. Mannheim, Germany: ZUMA.

Alexa, M., Rostek, L. (1996). 'Computer-assisted, corpus-based analysis text with TATOE', pp. 11-17 in *ALLC-ACH96, Book of Abstracts*. Bergen, Norway, University of Bergen.

Alexa, M., Zuell, C. (in press). Software for computer-assisted text analysis: a review. ZUMA Nachrichten Spezial. Mannheim, Germany: ZUMA.

Alexa, M., Schmidt, I. (in press): Modell einer mehrschichtigen Textannotation für die computerunterstützte Textanalyse. In Möhr, W., Schmidt, I. (eds) (in press): *SGML/XML - Anwendungen und Perspektiven.* Heidelberg: Springer-Verlag.

Carley, K. (1993). 'Coding choices for textual analysis: a comparison of content analysis and map analysis', *Sociological Methodology* 23: 75-126.

Carley, K., Palmquist M. (1992). 'Extracting, representing, and analyzing mental models', *Social Forces* 70: 601-636.

Evans, W. (1996). 'Computer-supported content analysis', *Social Science Computer Review* 114(3): 269-279.

Fielding, N.G., Lee, R.M. (1991). *Using computers in qualitative research.* Thousand Oaks, CA: Sage Publications.

Franzosi, R. (1990). 'Computer-assisted coding of textual data: An application to semantic grammars', *Sociological Methods and Research* 19: 225-257.

Franzosi, R. (1997). 'Labor unrest in the Italian service sector: An application of semantic grammars', pp. 131-145 in C.W. Roberts (ed.), . *Text analysis for the social sciences: methods for drawing statistical inferences from texts and transcripts.* Mahwah, N.J.: Lawrence Erlbaum Assoc. Publishers.

Hart, R.P. (1985). 'Systemic Analysis of Political Discourse: The Development of DICTION', pp. 97-134 in Sanders, K. (eds.), *Political Communication Yearbook: 1984.* Carbondale, IL: Southern Illinois University Press.

Huber, G.L. (1997). Analysis of Qualitative Data with AQUAD Five for Windows. Schwangau, Germany: Ingeborg Huber.

Klein, H. (1997a). 'Classification of Text Analysis Software', pp 355-362 in Klar, R., Oppitz, O. (eds)., *Classification and Knowledge Organization.* Proceedings of the 20th Annual Conference of the Gesellschaft fuer Klassifikation e.V. Heidelberg: Springer.

Klein, H. (1997b). INTEXT-Handbuch, Version 4.0. Jena, Germany: Mimeo.

Kuckartz, U. (1998). WinMAX: Scientific text analysis for the social sciences. User's guide. Berlin: BSS.

McTavish, D.G., Pirro E.B. (1990). 'Contextual content analysis', *Quality and Quantity* 24:245-265.

McTavish, D., Litkowski K.C. & Schrader S. (1997). 'A computer content analysis approach to measuring social distance in residential organizations for older people', *Social Science Computer Review*, 15 (2): 170-180.

Miller, G.A., Beckwith, R., Fellbaum, Chr., Cross, D., Miller, K., Tengi, R. (1993). Five Papers on WordNet™. CSL Report 43. Cognitive Science Laboratory. Princeton, NJ: Princeton University.

Mohler, P. Ph. & Zuell, C. (1998). TEXTPACK User's Guide. Mannheim: ZUMA.

Muhr, Th. (1996). 'Textinterpretation und Theorienentwicklung mit Atlas/ti', pp. 245-259 in Bos, W., Tarnai C. (eds.), *Computerunterstuetzte Inhaltsanalyse in den Empirischen Sozialwissenschaften.* Muenster: Waxmann.

Richards, L. (1998). NUD*IST Introductory Handbook. Victoria, Australia: QSR.

Roberts, C.W. (1989). 'Other than counting words: a linguistic approach to content analysis', *Social Forces* 68: 147-177.

Roberts, C.W. (ed.) (1997a). Text analysis for the social sciences: methods for drawing statistical inferences from texts and transcripts. Mahwah, N.J.: Lawrence Erlbaum Assoc. Publishers.

Roberts, C.W. (1997b). 'Semantic text analysis: on the structure of linguistic ambiguity on ordinary discourse', pp. 55-77 in Roberts, C.W. (ed*.), Text analysis for the social sciences: methods for drawing statistical inferences from texts and transcripts*. Mahwah, N.J.: Lawrence Erlbaum Assoc. Publishers.

Rostek, L., Alexa, M. (1998). 'Marking up in TATOE and exporting to SGML', *Computers and the Humanities* 31: 311-326.

Schrodt, Ph.A. (1998). KEDS: Kansas Event Data System. Version 0.9B7. Kansas University: http://www.ukans.edu/~keds/.

Sperberg-McQueen, M.C., Burnard, L. (eds.) (1994). Guidelines for the encoding and interchange of machine-readable texts (TEI P3). Chicago & Oxford, April 1994.

Stone, Ph.J. (1997). 'Thematic text analysis: New agendas for analysing text content', pp. 35-54 in Roberts, C.W. (ed*.), Text analysis for the social sciences: methods for drawing statistical inferences from texts and transcripts*. Mahwah, N.J.: Lawrence Erlbaum Assoc. Publishers.

Tesch, R. (1990). Qualitative research: Analysis types and software tools. New York: Falmer.

Tesch, R. (1991). ‘Software for qualitative researchers: analysis needs and program capabilities’, pp.16-37 in Fielding, N.G., Lee, R.M., Using computers in qualitative research. Thousand Oaks, CA: Sage Publications.

TextSmart™ 1.0 (1997). User’s guide. Chicago: SPSS Inc.

Weitzman, E.A., Miles, M.B. (1995). A software sourcebook: Computer programs for qualitative data analysis. Thousand Oaks, CA: Sage Publications.

Zuell, C. & Alexa, M. (1998). Ergebnisse der TEXTPACK Nutzerbefragung. Technischer Bericht. Mannheim, Germany: ZUMA.

Appendix 1

AQUAD http://www.aquad.com
Guenter Huber, University of Tuebingen, Germany.

AQUAD is a qualitative text analysis program which can be used for analysing interview transcripts, field notes, observation protocols, diaries, historical or political documents, etc. It supports category creation and manual coding on the basis of created categories, memo-writing, search-and-retrieval of words or coded lines and provides search algorithms for looking how two or more codes 'link' or co-occur in the same document.

ATLAS/ti http://www.atlasti.de/
Thomas Muhr, Berlin, Germany.

ATLAS.ti is a qualitative text analysis program supporting the researcher during the text interpretation process, particularly in managing text and coding categories, manual coding, memo-writing, exploring and comparing text and codes. The codes which the user defines and assigns to text segments can be linked in networks with a varying degree of complexity. ATLAS.ti supports the merging of different ATLAS.ti projects as well as teamwork, by allowing multi-authoring. The program provides for the HTML export of complete projects and generates a syntax file for further statistical processing of the coding with SPSS.

CoAn http://gwdu19.gwdg.de/~mromppe/soft/soft.htm
Matthias Romppel, University of Goettingen, Germany.

CoAn is a content analysis program. Based on a dictionary file CoAn will code texts both automatically and interactively. The program provides frequency of occurrence word lists, KWIC displays for all the coded words of a text, word comparison lists for two different texts and allows the user to export the coding to an SPSS readable file.

Code-A-Text http://www.codeatext.u-net.com/
Alan Cartwright, University of Kent at Canterbury, UK.

Code-A-Text is a program for the analysis of psychotherapeutic conversations as well as other in-depth dialogue texts, process notes, and answers to open-ended survey questions. Some combination of qualitative and quantitative analysis methodology is also supported. Texts to be analysed by Code-A-text can be ASCII or RTF files and they can be edited during analysis. Coding in Code-A-Text can be performed both automatically and interactively and can be based not only on a dictionary of words and phrases but, also, on user-defined nominal or ordinal scales. Code-A-Text provides rich search and retrieval support and as well as basic word counts for whole texts or selected sections. Codes can be converted to a numeric format and written into a command file that can be read by SPSS.

DICTION http://www.scolari.com/diction/Diction.htm
Roderick Hart, University of Texas, USA.

DICTION has been created specifically for the analysis and automatic coding of political discourse, such as presidential speeches. It attempts to determine the language characteristics of texts on the basis of calculated scores for five general categories, i.e. Activity, Certainty, Commonality, Optimism and Realism, comprising twenty-five sub-categories. DICTION uses dictionaries consisting of lists of characteristic words for each sub-category.

DIMAP-MCCA http://www.clres.com/
Ken Litkowski, CL Research, Gaithersburg, MD, USA/
Don McTavish, University of Minnesota, MN, USA.

DIMAP (Dictionary Maintenance Programs) is a dictionary development software which has an integrated content analysis component, the Minnesota Context Content Analysis (MCCA). Input in MCCA are English verbatim transcripts of speech or interviews, answers to open-ended questions, or other free textual material. MCCA supports automatic content analysis based on a dictionary comprising of 116 categories. The differentiating aspect of content analysis with MCCA is that the automatic categorisation (coding) per se is not the required analysis outcome, but rather a by-product of the analysis. The user of MCCA aims at obtaining *emphasis-scores* and *context-scores* for a given text corpus, which can be further statistically analysed and which are indicative of the content of the text or texts analysed.

HyperRESEARCH http://www.researchware.com/
ResearchWare, Randolph, MA, USA.

HyperRESEARCH is a qualitative text analysis program for both Windows and Macintoshes. It supports the definition of codes and manual coding of text segments according to one or more than one codes. The most sophisticated feature of HyperRESEARCH is its theory-building and testing module, which supports searches for a combination of codes using the Boolean AND, OR and NOT operators and retrieving cases, i.e. texts with coded test segments, with the specified combination of codes.

KEDS http://www.ukans.edu/~keds/
Philip Schrodt, University of Kansas, USA.

KEDS is a freely-available Mac-based program which has been specifically developed for the computer-assisted or fully-automated coding of international event data, contained in English lead sentences of wire service reports (Reuters news service) or in chronologies. KEDS relies on sparse and shallow parsing of the sentences, rather than using full syntactical analysis. The accuracy and robustness of KEDS in analysing English sentences relies on exploiting text type and domain specific information of the source texts: sentences found in English newswire reports have a relatively consistent structure.

NUD*IST http://www.qsr.com.au/software/n4/n4.htm
QSR Software, La Trobe University, Australia.

NUD*IST is a qualitative analysis program for both Windows and Macintoshes for the analysis of in-depth interviews, open-ended answers to surveys, group discussions, historical documents, etc. The user of NUD*IST can store, manage and code texts, search and retrieve both text and codes, write memos on documents and codes, and generate reports on the documents analysed. NUD*IST supports the coding of *text segments as well as documents* according to a coding scheme whose codes are may be nodes of an Index Tree. NUD*IST exports part or all of the *Index Tree* together with nominated details for further processing by the Inspiration or Decision Explorer software. Factual or demographic data can be imported and used as additional information for the analysis of texts. NUD*IST provides a utility for the merging of two or more NUD*IST projects as well as the exchange of projects between the Windows and Macs.

QED http://www.trimm.nl/qed/
TriMM MultiMedia, Holland.

QED is a qualitative text analysis program which can be used for the manual coding of text, for storing, searching and retrieving texts as set of texts, creating codes, assigning codes to texts or

text segments and searching for codes. All sorts of qualitative data, such as interviews, survey data, observation data, archive material, protocols or transcribed conversations can be analysed using QED.

TATOE http://www.zuma-mannheim.de/alexa/en/tatoe.htm
Melina Alexa, ZUMA, Mannheim
Lothar Rostek, GMD-IPSI, Darmstadt, Germany.

TATOE is a general text analysis tool, combining qualitative and quantitative functionality. It supports the manual or automatic coding of a text or a corpus of texts. The user can define and use one or more than one categorisation schemes for coding and the categories of each scheme may be hierarchically organised. Texts which are already coded may be imported in TATOE and their coding is stored as a separate categorisation scheme. TATOE supports text and coding exploration, e.g. word and code searches, frequencies of occurrences, KWIC and network displays. An interesting feature of TATOE is the possibility to define search patterns consisting of a combinations of codes and strings and code the matching instances. TATOE exports coding as an XML file and to SPSS for further processing.

TEXTPACK http://www.zuma-mannheim.de/software/en/textpack/
ZUMA, Mannheim, Germany.

TEXTPACK is a program for content analysis. It supports the automatic coding of any type of text, such as open-ended questions, interviews, newspaper articles, etc. Automatic coding in TEXTPACK is based on dictionaries provided by the user. The program supports text exploration by providing word frequencies, cross- reference lists, KWIC lists, comparisons of the vocabulary of two files, supporting in this way the development and validation of the dictionary used for coding. The coding can be exported for further statistical processing with SPSS and SAS.

TextSmart http://www.spss.com/software/textsmart/
SPSS Inc., Chicago, USA.

TextSmart is a software for the automatic as well as interactive coding of answers to open-ended survey questions. It uses linguistic technology, like word stemming, and statistical algorithms (clustering and multidimensional scaling) for generating automatically ‘categories’ and coding survey responses. TextSmart is “dictionary-free”, in that there is no need to create a coding scheme or dictionary before running the analysis, i.e. before coding. The user may also create her own categories simply by specifying which words or combinations of words belong to a category. In both automatic or interactive creation of categories, codes are automatically assigned to the texts. Demographic data can also be used during analysis and the coding can be exported for further processing with SPSS.

WinMaxPro http://www.winmax.de/
Udo Kuckartz, Berlin, Germany.

WinMAXpro is a program for qualitative-oriented text analysis. It can be used for the analysis of different types of texts, such as interviews, case histories, field notes, answers to open-ended questions, letters, etc. The program supports manual coding, the construction and maintenance of a hierarchical categorisation scheme, memo writing, the definition and usage of case variables or demographic information, and the exploration of coding by means of logical operators. It, also, calculates frequencies of occurrence of words or strings of words and codes. WinMAXpro supports teamwork and enables the merging of WinMAXpro projects. The export the coding for further statistical processing with SPSS is supported.

WordStat http://www.simstat.com/

Normand Peladeau, Provalis Research, Montreal, Canada.

WordStat is a content analysis module of the statistics program SIMSTAT. WordStat can be used for the automatic as well as interactive coding mainly of answers to open-ended questions, but, also, of other typical short textual data such as, for instance, short advertisements. WordStat's coding functionality is dictionary-based: the automatic coding uses already existing or new user-defined dictionaries. The user can, also, code interactively by means of assigning unique keywords (codes) anywhere in a text. Since WordStat is fully-integrated with SIMSTAT it supports the user in assessing the reliability of coding by providing eight different inter-raters agreement measures. WordStat offers general text statistics information, such as word counts, category counts or calculation of word by word co-occurrence matrices, and provides KWIC as well as Code-In-Context displays. The coding performed with WordStat can be, naturally, further statistically analysed with SIMSTAT. In addition, the WordStat dBase output files can be input to SPSS.