Software for Content Analysis – A Review

Will Lowe
wlowe@latte.harvard.edu

1 Introduction

Software for content analysis divides, according to its intended function, into three major categories. The first set of programs perform *dictionary-based content analysis*. They have the 'basic handful' of text analysis functions, involving word counting, sorting, and simple statistical tests. The basic handful are described in the next section. The second set contains *development environments*. These programs are designed to partially automate the construction of dictionaries, grammars, and other text analysis tools, rather than being analyzers themselves. Development environments are more similar to high-level text-specific programming languages than to freestanding content analysis packages. The third category contains *annotation aids*. While an annotation aid can often perform some automatic content analysis, it is intended more as an electronic version of the set of marginal notes, cross-references and notepad jottings that a researcher will generate when analyzing a set of texts by hand.

The next section describes the basic handful of text analysis functions, and the rest of the paper provides brief descriptions of twenty-one content analysis programs. Some recommendations are made in the conclusion.

1.1 The Basic Handful

The basic handful of functions consists of word frequency counts and analysis, category frequency counts and analysis, and visualization.

Word Frequency Analysis

Word frequency analysis provides a list of all the words that occur in a text and the number of times they occur. More sophisticated methods split the text into subparts, e.g. chapters, and create frequency lists for each part. Lists can be compared either visually, or using a statistical test such as χ^2, to see if their are significantly more mentions of particular words in one part than another. Another common use for the subpart procedure is to compare different sources addressing the same substantive question to measure how different their treatment of it is on the basis of the sorts of words they use.

Statistically this procedure can sometimes be reasonable because the counts from one source are compared with the total counts for all words over all the sources; significant differences may then track differences of emphasis across sources[1]. Some packages make use of synonym lists or lemmatize before the analysis in order to merge word counts. Lemmatization removes the grammatical structure from the surface form of a word, leaving only the stem; words are then counted as identical when they share a stem. For example, a lemmatizing frequency count would treat 'steal' and 'stole' as the same word. Lists of lemma and synonyms are naturally language specific.

Word frequency analysis is the simplest form of content analysis. In fact most operating systems (e.g. Unix/Linux, Mac OSX, and recent versions of Windows) have utilities to perform basic word counting and sorting built in.

Category Frequency Analysis

Content analysis programs almost all allow the user to specify a dictionary. 'Dictionary' in this context means a mapping a set of words or phrases to one word; the one word is the label of a substantive category and the set describes the words or phrases that indicate the tokening of the category in text.

As an example, the Linguistic Inquiry and Word Count (LIWC) dictionary maps the word set {*ashes*, *burial**, *buried*, *bury*, *casket**, *cemet**, *coffin**, *cremat**, *dead death**, *decay**, *decease**, *deteriorat**, *die*, *died*, *dies*, *drown**, *dying fatal*, *funeral**, *grave**, *grief*, *griev**, *kill**, *mortal**, *mourn**, *murder* suicid**, *terminat**} to LIWC category 59, `death`. The asterisks are 'wild-card' characters telling the program to treat 'cremating', 'cremated' and 'cremate', as all matching *cremat**, and thus all mapping to category 59.

Category counts allow a slightly more sophisticated analysis because they allow the user to provide a more explicit model of latent content in text. The implicit model of text generation is that the author of the text has some message expressed in terms of categories, and that this message is 'coded' into English when she writes. Coding entails picking any one of a set of English words that represent that concept, perhaps constrained by grammatical or pragmatic criteria. If the content analyst can recover or construct the word set used by the author, it can be placed in a dictionary and used to decode other texts.

According to the LIWC scheme the sentence "Her suicide caused him to consider his own mortality" refers to the categories of 'death' and 'metaphysics' twice, 'social' three times, and 'causation' once:

> Her–SOCIAL suicide–DEATH/METAPH caused–CAUSE him–SOCIAL to consider–COGMECH his–SOCIAL own mortality–DEATH/METAPH.

But according to the implicit model of LIWC, "He thought of his own death only because she killed herself" is an equally good instantiation of the underlying content because it tokens the same categories the same number of times. Of course many other sentences them these categories too, and many of the are quite unrelated in meaning.

When a text is reduced to its category tokens with respect to some dictionary the same statistical analysis can be performed as with word counts. For most applications of automated content analysis, a word is re-

[1] On the other hand, they probably also track differences in writing style and other extraneous factors.

duced to a vector of category counts. Different texts can be compared either across within each category, or more usefully, by looking at high-dimensional distance measures between the complete vectors associated with each text. Most information retrieval programs, e.g. Google, will make use of a similar vector representation of texts – each query is converted into a sparse category vector by coding it as if it were a very short text, and this vector is compared geometrically to all available other vectors to find the nearest, that is, most relevant text to the query.

Visualization

When a text has been reduced to vector form, either by counting words or categories, it can be visualized. Two standard methods provided by most content analysis programs are clustering and multidimensional scaling. Cluster analysis is no doubt familiar, but the multidimensional scaling bears some discussion. It appears that most scaling procedures packaged for content analysis perform metric rather than non-metric multidimensional scaling. This means that the programs are looking for the linear mapping (for visualization purposes it will be a plane) that passes through the vectors and captures most variation in their positions when they are projected onto it. Metric methods therefore enforce linear structure, which may or may not be reasonable. More computationally intensive methods are non-metric, and consider not the positions of the vectors but their distance ranking to one another. Non-metric methods attempt to preserve ranked distances in their mapping to the plane, and thus allow more non-linear structure to appear in the final visualization.

Why does this difference matter? It might appear that visualization functions are an advantage in a content analysis program, and this is may be true for preliminary data exploration. But researchers will most likely end up putting their data into a regular statistics package at some point, perhaps to get a more sophisticated statistical analysis. Since most modern statistics packages have very sophisticated visualization functions, the visualization will almost certainly be better performed then. This will also be desirable in the case where the content analysis package does not (or will not) document the exact clustering or visualization routine being performed.

Other Basic Functions

Several programs can generate concordances, sometimes described as KWIC (’key word in context’) analysis. The table below is a selection of lines from a small window full concordance for the word ‘content’ in the paragraphs preceding this one.

—	—	Software	for	**content**	analysis	divides	according	to
can	perform	dictionary	based	**content**	analysis	They	have	the
often	perform	some	automatic	**content**	analysis	it	is	intended

Although computing concordances is not really a method of automated content analysis, it can be a very fruitful way to examine the data in the process of designing a content analysis; one example use for concordance analysis would be to quickly discover, without having to read the entire text, that the presence of a particular word occurs only in a subset of its possible substantive roles, even when we might expect it to be more broadly distributed on purely linguistic grounds (e.g. that taxes are only mentioned when the text is talking about lowering them.)

Concordances are also useful representation for discovering sets of words that co-occur reliably with the keyword, and thus might be natural choices for dictionary word sets.

Finally, with the addition of some minor annotation capability the researcher may manually code each instance as being of a particular category, either as part of a 'training set' for subsequent automated analysis, or simply as quick confirmation that, say 75% of mentions are of a particular type. The principle advantage of concordances in all these roles is that they lighten the reading burden of the researcher, so she can work with a larger volume of text.

2 Content Analysis Programs

This section describes twenty-one content analysis packages. They are divided into dictionary-based programs and development environments. A final section describes the two most popular annotation aids. Where possible each section states the platforms that the software runs on, the licensing scheme, the accessibility of the code-base and whether it is able to work with non-English language text.

Licensing cost has been distinguished from the accessibility of the code-base because although many packages are free to use, their code is not available. Being able to see the code is useful if one needs to know exactly what is going on when the program performs more complex analysis. In this respect the software is effectively proprietary. However, since there is no tradition among Windows and Mac users to make their code available even when the software is written to be given away, it may only be convention that makes the code-base inaccessible. That is, individual authors of free software may happily provide code details on request. This will certainly not be the case for the commercial packages.

2.1 Dictionary-based Content Analysis

CATPAC

—Homepage: `http://www.terraresearch.com/catpac.cfm`

—Operating Systems: Windows

—License:

Commercial $595
Academic $295
Student $49

—**Code base:** Proprietary (executable only)

—**Languages:** English (ASCII only)

Despite the bold claims of the manufacturer:

> "CATPAC is an intelligent program that can read any text and summarize its main ideas. It needs no pre-coding and makes no linguistic assumptions."

CATAC performs only the basic handful of functions. Visualization involves cluster analysis and multidimensional scaling. Cluster analysis can be interactive. CATPAC also apparently allows three dimensional visualizations with appropriately colored glasses.

CATPAC seems adequate to the basic handful. However the user interface is weak and the `http://www.galileoco.com/pdf/catman.pdf` is atrocious.

Computer Programs for Text Analysis

—**Homepage:** `http://www.dsu.edu/~johnsone/ericpgms.html`

—**Operating Systems:** MS-DOS

—**License:** Freeware

—**Codebase:** Proprietary (executable only)

—**Languages:** English (ASCII only)

These are a set of utility programs run from the DOS command line. They cover the basic handful except for visualization, and are designed primarily for literary analysis.

Concordance

—**Homepage:** `http://www.rjcw.freeserve.co.uk`

—**Operating Systems:** Windows

—**License:** $89 + $10 handling fee. $40 per subsequent license.

—**Codebase:** Proprietary (executable only)

—**Languages:** English, Chinese (See `http://deall.ohio-state.edu/chan.9/conc/concordance.htm`)

Concordance is marketed as a way of producing and publishing concordances for literary texts (See for example http://www.dundee.ac.uk/english/wics/wics.htm) However the program also performs a superset of the basic handful of word analysis and category analysis functions, including regular expressions and lemmatization. (Lemmatization involves reducing all instances of a word to its stem.) There appears to be no visualization option.

The most appealing aspect of Concordance is its potential for processing text in languages other than English (see http://deall.ohio-state.edu/chan.9/conc/concordance.htm for more detail). It is not clear from the manufacturer's information whether reason Concordance can deal with Chinese is because it processes all text in Unicode, or because it has been specifically designed for Chinese scripts. If the underlying processing model uses Unicode then it is reasonable to expect support for other languages. If, on the other hand, it is an ad-hoc extension then Concordance is likely to be less generally useful.

Diction

—Homepage: http://www.sagepub.com

—Operating Systems: Windows

—License:
Commercial $189
Academic $129

—Codebase: Proprietary (executable only)

—Languages: English (ASCII only)

Diction uses built -in dictionaries to code for:

> "*Certainty* - Language indicating resoluteness, inflexibility, and completeness and a tendency to speak ex-cathedra. *Activity* - Language featuring movement, change, the implementation of ideas and the avoidance of inertia, *Optimism* - Language endorsing some person, group, concept or event or highlighting their positive entailments, *Commonality* - Language highlighting the agreed-upon values of a group and rejecting idiosyncratic modes of engagement, *Realism* - Language describing tangible, immediate, recognizable matters that affect people's everyday lives."

Category counts using built-in dictionaries are apparently 'standardized'. However it is not possible to say what this means since the dictionaries and standardization procedure appear to be proprietary. Custom dictionary construction is possible.

General Inquirer

—Homepage: http://www.wjh.harvard.edu/˜inquirer/

—Operating Systems: Any (Java program)

—License: Free for academic use

—Codebase: Proprietary(?)

—Languages: English (ASCII only)

The General Inquirer performs the basic handful of functions, without visualization. The program has 182 built-in categories that are the result of merging several existing content analysis dictionaries (See http://www.wjh.harvard.edu/˜inquirer/homecat.htm for details). Some dictionaries are reduced in detail inside the program. For example original Osgood semantic differentials are real valued factor loadings, whereas the versions in Inquirer have only 0-1 variables for each factor.

Custom dictionary construction is not possible, but not straightforward. The Inquirer's author has recommended against it (pers. comm.).

HAMLET

—Homepage: http://www.apb.cwc.net/homepage.htm

—Operating Systems: MS-DOS

—License: Free "for personal use"

—Codebase: Proprietary (executable only)

—Languages: English (ASCII only)

HAMLET computes word co-occurrence counts, maps co-occurrences into similarity matrices and performs cluster analysis or multidimensional scaling on the result. This is a small subset of the basic handful of functions.

T-LAB

—Homepage: http://www.tlab.it

—Operating Systems: Windows

—License: $520 single user license

—Codebase: Proprietary (executable only)

—**Languages:** English, Spanish and Italian.

T-LAB performs the basic handful of functions in one of three languages. Adding a language increases the price.

WinATA

—**Homepage:** http://www-users.aston.ac.uk/˜roepj/guide/guide.htm

—**Operating Systems:** Windows

—**License:** Free

—**Codebase:** Proprietary (executable only)

—**Languages:** English (ascii only)

WinATA performs the usual handful of functions with some small variations. It has a small user base of corpus linguists in the UK.

TEXTPACK

—**Homepage:** http://www.social-science-gesis.de/en/software/textpack/index.htm

—**Operating Systems:** Windows

—**License:**
Commercial single user E300
Student E100
Network E1500

—**Codebase:** Proprietary (executable only)

—**Languages:** English (ascii only)

TEXTPACK performs the basic handful of functions, except visualization. A recent political science application of TEXTPACK has been to automate the coding of political manifestos (Laver and Garry, 2000).

LIWC

—**Homepage:** http://www.erlbaum.com

—**Operating Systems:** Windows, Mac

—**License:** Single user $99

—Codebase: Proprietary (executable only)

—Languages: English (ascii only)

LIWC (Linguistic Inquiry and Word Count) performs the basic handful of functions. The software is designed around a custom dictionary created by James Pennebacker, but user dictionaries can be used.

MonoConc / ParaConc

—Homepage: `http://www.ruf.rice.edu/˜barlow/mono.html`

—Operating Systems: Windows

—License: Free

—Codebase: Proprietary (executable only)

—Languages: English (ascii only)

MonoConc is intended as a concordance generation package. ParaConc performs the same function on aligned corpora, e.g. the Hansard corpus of Canadian government debate transcripts containing sentence-by-sentence aligned French and English translations. Both packages perform the basic handful, except for visualization.

Lexa

—Homepage: `http://nora.hd.uib.no/lexainf.html`

—Operating Systems: Windows

—License: Free

—Codebase: Proprietary (executable only)

—Languages: English (ascii only)

Lexa performs the basic handful, except visualization. It also lemmatizes. The authors claim that linguists are the target audience.

SPSS TextSmart

—Homepage: `http://www.spss.com/textsmart/`

—Operating Systems: Windows

—License: Unknown

—Codebase: Proprietary (executable only)

—Languages: English (ascii only)

SPSS's TextSmart is a colorful interface to the basic handful. It also deal with synonyms and provides a dictionary-free categorization method based on co-occurrence. This is proprietary so it is not possible to say how appropriate it will be to any particular content analysis.

SPSS have redesigned their website and lost all but a handful of links to TextSmart. In particular it is not possible to tell what the pricing scheme is.

VBPro

—Homepage: `http://excellent.com.utk.edu/˜mmmiller/vbpro.htm`

—Operating Systems: MS-DOS

—License: Free

—Codebase: Proprietary (executable only)

—Languages: English (ascii only)

VBPro is considered among quantitative content analysts to be the baseline package. It performs the basic handful of functions, except visualization, but the author provides another free package to cover this. The interface is relatively straightforward though extremely spartan.

A political science application of VBPro is described in Denham and Miller's on newspaper coverage of presidential campaigns (Denham and Miller, 1993).

WordStat

—Homepage: `http://www.simstat.com/wordstat.htm`

—Operating Systems: Windows

—License: $278 ($129 + required Simstat base $149)

—Codebase: Proprietary (executable only)

—Languages: English, French, Spanish and Dutch

WordStat is a module for the SimStat, a bare-bones statistics package resembling SPSS. SimStat performs the basic handful and has some annotation capability.

2.2 Development Environments

Profiler Plus

—**Homepage:** http://www.socialscienceautomation.com

—**Operating Systems:** Windows

—**License:**

Commercial $770, $400 p.a. thereafter
Academic $385, $200 p.a. thereafter
Lite (Limited functionality) $60

—**Codebase:** Proprietary (executable only)

—**Languages:** English (ASCII only)

Profiler Plus is a mixture of development environment and content analysis program. All functionality appears to be proprietary. Various add-on packages allow latent trait analysis (See http://www.socialscienceautomation.com/Lta.pdf for a high-level description), cognitive mapping and operational code analysis. It may be possible to discover what these operations involve by contacting the author directly. The interface for Profiler Plus is not very intuitive, and documents are limited to ASCII text.

DIMAP

—**Homepage:** http://www.clres.com

—**Operating Systems:** Windows

—**License:**

Commercial $330
Academic (See below)

—**Codebase:** Proprietary (except perhaps for the Franklin parser)

—**Languages:** English (ASCII only)

DIMAP is an extremely rich development environment for creating dictionaries for text analysis. The program author is a computational linguist, and the software has been used in academic competitions e.g. SENSEVAL, a word sense disambiguation benchmark. This ought to make it easier to find out what algorithms DIMAP uses. DIMAP is aimed principally at computational linguists interested in information extraction (see also KEDS below) and lexicography, and less at standard social science content analysis. However, DIMAP has a content analysis package module available: ‘Minnesota Contextual Content Analysis’ (MCCA).

MCCA performs the basic handful of functions, and has a built-in dictionary. The built-in dictionary can be modified. The dictionary is described in a useful paper on developing category systems for content analysis: http://www.clres.com/catprocs.html and in Litowski's paper in Social Science Computing Review (Litowski, 1997).

The academic licensing agreement for DIMAP appears to give away the software at no cost. In return the licensee is obliged to turn over all her work that uses DIMAP. Here is an excerpt from the license. Italics are mine:

> "3.2 *The University will complete and deliver to CL Research* (a) bug report forms provided by CL Research, (b) *copies of any dictionaries developed for research purposes by students or faculty using* DIMAP*, and* (c) *descriptions of such dictionaries (if developed by students or faculty in their coursework or research)*. The University agrees that, in any publications for which supporting dictionaries created using DIMAP, acknowledgment of the use of DIMAP will be made."

Visual Text

—Homepage: http://www.textanalysis.com

—Operating Systems: Windows

—License: (Contact company)

—Codebase: Proprietary

—Languages: English

VisualText is a rich development environment for building text analyzers. It includes a text analysis language similar to C++ for programming. All analysis methods, including details of the language implementation, appear to be proprietary. Like DIMAP, and to a lesser extent Profiler Plus, VisualText is aimed at those interested in information extraction rather than traditional social science content analysis. However, VisualText performs a large superset of the basic handful of functions.

KEDS / TABARI

—Homepage: http://www.ku.edu/˜keds/

—Operating Systems: MS-DOS, Mac, Unix/Linux

—License: Free

—Codebase: Open source

—Languages: English (ascii only)

KEDS/TABARI is an open source code package geared to information extraction from news leads. KEDS is the name of the Mac and Windows version and TABARI is the version for Linux or Unix. The Mac version of this software has a graphical interface, the others are DOS-style text menu interfaces. However the underlying code is the same.

KEDS is designed to populate a database with actors and events according to a predetermined event ontology. Originally the ontology was limited to the McClelland's World Events Interaction Survey codes (McClelland, 1978), and a more detailed scheme developed by Doug Bond's group (Bond et al., 1997), but Schrodt's group has recently added a new coding focusing on mediation events (Gerner et al., 2002). None of these event schemes are obviously appropriate to content analysis because they attempt to abstract away from the tone and word-choice decisions made by the document author in order to uncover the event information being expressed. Since it is precisely these decisions that content analysts assume to be revealing about the latent content of a text a new coding scheme would have to be developed from scratch. On the other hand, developing such a scheme has considerable potential, at least if sufficient time and man-power is available.

There are two reasons for this: First, it may not be necessary to alter the program code at all to allow a new coding scheme; the coding scheme representation is contained, not in the parser, but in a separate dictionary containing mappings from a regular expression-like patterns centered on verbs, to event categories. Dictionary construction has always been a task for the users of the program because it involves significant domain knowledge and depends on the sort of text KEDS will have to deal with. Schrodt has emphasized that this process is relatively straightforward (though tedious) and that it has been performed many times by different researchers.

The second reason that adaptation of KEDS for content analysis may have potential is that the parser provides 'for free' a level of linguistic analysis that is absent from most other content programs. Clearly, knowing the source and target actors, that is the grammatical subject and object in an accusation allows a much more detailed analysis, perhaps in terms of operational codes or other explicitly psychological constructions, than simply knowing that an accusation has been made and that accusation fall into one or more dictionary categories. Of course the reason that other programs typically do not provide this level of analysis is that in the absence of a strong computer science education it is hard to write a parser, and even with the requisite background it is hard to write a good one.

Neither of these reasons depends very closely on the parser being KEDS rather than some other choice. The important feature of a suitable program is that the dictionaries its parser uses must be separable from the main program code so researchers are not required to know anything about how the parsing process is implemented. This separation is clearer in KEDS than in the other development environments described in this document, mostly because the other environments are intended to help computational linguists, not social scientists. It is possible that an alternative parser would be able to perform the same role as KEDS, but I have not found one with the necessary separation.

One way to assess the potential of an adapted KEDS for content analysis would be to take an existing dictionary, for example one coded by Schrodt that covers some area of interest, and replace the event codes associated with verb structures mentioned in the dictionary with 'content' codes. The performance of KEDS over a test text ought to give some indication of whether adaptation is a viable option.

The final consideration for using KEDS is that of language. KEDS can only English language text; any other language would require a completely redesigned parser. Naturally this will be true of any other alternatives to KEDS, and it is the price of a potentially more sophisticated linguistic analysis. One possible treatment of this problem is to attempt to abstract the mapping of verb structures onto content categories away from KEDS, expressing it in a format that would be consistent with a wider variety of parser application interfaces. If the KEDS regular expression-like dictionary representation were kept then a more standard piece of content analysis software might well be able to perform some of the same analysis.

2.3 Annotation Aids

Atlas-ti

—**Homepage:** http://www.atlasti.de

—**Operating Systems:** Windows, MS-DOS

—**License:** $250

—**Codebase:** Proprietary (executable only)

—**Languages:** English only(?)

Atlas-ti is an annotation and note-keeping aid that has some limited automatic content analysis functionality. The program is highly developed and extremely professionally constructed. For example, Atlas exports to HTML and claims to be able to import and export XML documents. It has a wide variety of annotation styles including user-defined map structures, and can be fully cross-referenced.

XML capability suggests that Atlas will be able to deal with multilingual text – indeed the list of frequently asked question on the manufacturer's homepage claims that working with Hebrew is possible. However, neither I not Jason Lyall have been able to persuade Atlas to accept Cyrillic text.

For doing manual research Atlas seems useful and is very well thought of among qualitative researchers (See for example Tom Wilson's review for Information Research (Wilson, 2001).) However it is not clear that it is really an *automated* content analysis tool (similar observations can be made about NUD*IST). For automated content analysis, the main advantage of Atlas would seem to be as a benchmark for automated coding output: The researcher manually codes a set of documents with a predefined coding scheme, an automated coder processes the same documents, and the results are compared.

NUDIST

—**Homepage:** http://www.qsr-software.com

—**Operating Systems:** Windows

—License:
Single user $325
2-30 licenses $260

—Codebase: Proprietary (executable only)

—Languages: English only

NUD*IST is has very similar functionality to Atlas-ti, with a slightly less smooth interface and a more constrained set of annotation structures. Atlas-ti and NUD*IST are compared in more detail by Christine Barry's review for Sociological Research Online (Barry, 1998)

3 Choice Criteria

The variation among programs can be described in terms of a handful of variables, obtained by asking:

1. How complex an analysis can it perform?

2. Can it run on languages other than english?

3. Is the code base (or dictionary) proprietary?

4. Is there an established user base?

5. Does it only run on Windows?

3.1 Complexity and Language Constraint

To some extent the first two criteria trade-off; to perform the basic handful it is not necessary for a program to know anything about the structure of any language, only that it be able to manipulate all the characters and determine where words begin and end. Most programs have trouble with the latter requirement, though it would be fairly straightforward to write something with the capabilities of, say VBPRO, that would solve this problem. Dictionaries are in principle no more difficult to write in languages other than english, though it would not be easy to work from existing ones.

In contrast, complex analyses such as cognitive mapping, or anything else requiring the ability to parse text, will not generalise across languages. If we were to use KEDS as the basis for a complex content analysis, and wished to extend the same analysis to German, an entirely new system would have to be acquired (or written). This holds despite the relative ease with which the progams involved could be altered to deal with umlauts and the ß. The Profiler Plus is in the same position, although the source is proprietary so we would not be able to alter the basic algorithms.

3.2 Proprietary Methods

If SPSS provided a new non-linear regression algorithm but refused to specify how it was computed, most academic researchers would not be comfortable using it. It seems to me that if we are to bring textual analysis into the social scientist's standard quantitative toolkit alongside t-tests and logit analysis, then it is important that the algorithms and models underlying the analysis be as open as any regression model. Naturally this is an 'in-principle' argument since if most users of statistics packages do not know the details of every model and algorithm in theor package. If they did they would probably not need the packages in the first place. But it does seem important at least to be able to find out what sort of processing is occurring inside the program, and to be able to specify it in publications.

For programs performing the basic handful it is less important to have access the processing detail, since it amounts to word spotting and counting, but it is important to have access to the dictionary. For example, Diction and LIWC appear not to allow access to their own dictionaries, so no researcher can be sure just what they are coding. Similarly programs that claim to be based on co-occurrence analysis rather than dictionaries, such as SPSS TextSmart and CATPAC, ought to fully specify how co-occurrences are computed, though they do not.

For programs attempting more complex analysis such as Profiler Plus and WorldView, knowing the processing details is more important. Although the user is expected to construct her own dictionary, and its application is relatively transparent, it is still not clear how World View manipulates the coded data into a cognitive map.

It is important to note that the question of proprietary methods applies equally whether the software is sold or given away. It is no consolation that a program is free, if is is also completely opaque.

Annotation Aids

In contrast to the programs performing the basic handful and development environments, there is much less cause to have an open codebase and accessible dictionaries in the annotation aids. The reason is that these programs are not primarily intended as inference devices, but more like collations of research notes. If no academic conclusions rest on a formal analysis directly from the program it is of no importance to know exactly how they work.

Naturally, in the absence of a standardized exchange format for research notes – and it is unclear what that might look like, even in theory – the propriety of the methods used would require that two researchers working on the same problem both have the program installed, but that is not different to the requirement that two researchers have similar versions of Microsoft Word when they send each other drafts.

Alteration

The presence of proprietary methods in a program is also important if we expect to have to alter it in any way. One of the advantages of KEDS over Profiler Plus is that it is possible to add more functionality for particular research applications. This may be important even for programs performing the basic handful, if more different counting methods, or other language support is wanted.

3.3 Licensing Issues and User Base

The programs described above are either free or have a standard commercial license, often discounted for academic use. The only questionable arrangement is that of DIMAP: It seems to me ethically unsound that users of DIMAP should be required under an academic license to turn over their dictionary construction work to the program manufacturers. This licensing position stands in stark contrast to that of all manufacturers of proprietary statistical packages e.g. Gauss and S-Plus, who encourage users to make their functions freely available, and provide links to them.

For analyses of any complexity it is useful to be able to ask for help or advice, either from the author of the program, or from an existing community of fellow users. Many of the programs above have extremely small user bases. The notable exceptions are Atlas-ti, NUD*IST, and perhaps Concordance. And although the user bases for KEDS and Profiler Plus are not large, we do have relatively easy access to the authors. This is probably less important for KEDS since the source code is available.

3.4 Platforms

Eighteen out of twenty-one of the programs run only on Windows (three only on MS-DOS). LIWC runs on Windows and Macs, and the General Inquirer and KEDS run on Windows, Mac and Unix/Linux. This reflects the worldwide majority of Windows users, and need not be a problem in itself, despite still considerable numbers of Mac users across the social sciences. Consequently operating system is not a strong criterion for choosing packages – it'll almost certainly require Windows.

4 Conclusions

The programs described here suggest two broad lines of research. In the first, programs performing relatively complex content analyses are used to study english language texts. In the second, texts in various languages are analysed using dictionary and co-occurrence based methods. Naturally both may be pursued at the same time. Programs in either line may be augmented by code I write to perform analyses of particular interest that are not provided in the programs.

For the first line of research, my preference is towards using KEDS rather than Profiler Plus or Visual Text. The less important reason for this is that KEDS is open source, free and I have some experience with it. The more important reason is that it has a large set of dictionaries already developed for another purpose (information extraction) that look like they can be re-labeled according to contentful criteria, or at least used to facilitate more dictionary construction.

For the second line of research, a natural choice would seem to be a combination of Concordance with either VBPro, WinATA, and/or something I write with the the same or increased functionality that deals with other languages. WordStat and TextPack seem expensive equivalents to VBPro or WinATA.

Differences among programs in the second line are mostly a matter of taste rather than functionality, so it is important where possible to try a demonstration version before coming to any conclusion.

Finally, although I would not recommend using either of the annotation aids for content analysis, these programs are potentially very useful for other purposes. Atlas-ti offers a demonstration version.

References

Barry, C. (1998). Choosing qualitative data analysis software: Atlas-ti and Nudist compared. *Sociological Research Online*, 3(3).

Bond, D., Jenkins, J. C., Taylor, C. L., and Schock, K. (1997). Mapping mass political conflict and civil society: Issues and prospects for the automated development of event data. *Journal of Conflict Resolution*, 41(4):553–579.

Denham, B. and Miller, M. M. (1993). Public opinion polls during the 1988 and 1992 presidential election campaigns: An analysis of horserace and issue coverage in prestige newspapers. Paper presented at the annual meeting of Midwest Association for Public Opinion Research.

Gerner, D. J., Abu-Jabr, Schrodt, P. A., and Yilmaz, O. (2002). Conflict and Mediation Event Observations (CAMEO): A new event data framework for the analysis of foreign policy interactions. Paper presented at the Annual Meeting of the International Studies Association.

Laver, M. and Garry, J. (2000). Estimating policy positions from political texts. *American Journal of Political Science*, 44(3):619–634.

Litowski, K. C. (1997). Category development based on semantic principles. *Social Science Computing Review*, 15.

McClelland, C. (1978). World event/interaction survey, 1966-1978. WEIS Codebook ICPSR 5211, Inter-university consortium for political and social research, University of Southern California.

Wilson, T. (2001). Review of Atlas-ti. *Information Research*, 6(3).