[TEXT, ANALYSIS, TOOLS].define()

G. ROCKWELL
Communication Studies and Multimedia, McMaster University

S. SINCLAIR
McMaster University

J. CHARTRAND
Open Sky Solutions, McMaster University

The unusual nomenclature of this paper's title is meant to draw attention to one of the conceptual features that has intrigued us the most in the development of the Text Analysis Portal for Research (TAPoR) : the simultaneous modularity and interdependence of the three substantives that describe our work of elaborating text analysis tools. To better understand what we are intuitively doing in developing the Portal (without always making our presuppositions explicit beforehand), and to imagine how the Portal can best fulfil its mandate as a workspace for scholars working with electronic texts and tools, we are motivated to examine text analysis tools both at the atomic and molecular level. Or, to return to the programming metaphor of the title, we wish to examine each object individually (Text, Analysis, Tools) and also as a composite object of objects (Text Analysis Tools).

The ambiguity of the method define from the title (whether it applies to the individual objects iteratively or to the collection of objects) is deliberate. The pseudo-code of this paper is polymorphous: the process of defining (that constitutes the content of the paper itself) operates on different levels and on several types of objects. Furthermore, it should be noted that the use of unquoted, capitalized words for the objects to define is deliberate: as per object-oriented convention is, these represent classes of things rather than particular instances. As such, we are not so much interested in, say, the use of tools to analyze a particular text, but rather, the particularities of texts in general as relevant to use of analysis tools. More pragmatically, we are interested in how the concepts that we take for granted in developing text analysis tools can in fact yield a rich array of useful design principles for the Portal, when examined more closely.

This paper is structured as a sequence of definitions of the relevant components in isolation, accompanied with - in increasing intensity - a discussion of how these component are transformed when combined with one another. In other words, text analysis tools are not merely the amalgam of its constituent parts, but some class of object that extends beyond them. We will conclude by outlining some of the practical consequences that these reflections might have on the next phases of developing the TAPoR Portal.

1. Text

At first glance the concept of text seems relatively easy to define, something like "a meaningful sequence of characters, or abstract symbols, that forms a structural unit." Debatable though this definition may be, it certainly allows us to identify an essential common characteristic that encompasses everything from Egyptian hieroglyphic tablets to the Gutenberg bible and even text messages that are exchanged through mobile phones. Just as importantly, it allows us to distinguish such objects from other human artefacts such as hammers, oral stories, and television shows, which do not use symbolic characters to transmit meaning within a defined scope.

However, several potential problems with this definition quickly become apparent. For instance, how do we define something as fluid and subjective as meaning? Similarly, how do we delineate as a text an object that may be structurally complex (cf. clauses, sentences, paragraphs, chapters, sections, books, volumes, etc. in prose). The latter assumes even greater significance since the structuralist and poststructuralist theorizing of intertextuality. As Roland Barthes reminds us, texts are themselves an interweaving of elements – from the etymological roots of the Latin textere for tissue – and those elements can include the most culturally diverse objects (not just other texts). This leads theorists such as Julia Kristeva to state that everything, including culture itself, is a text. Although this logical expansion of the term text is theoretically generative, such a move also ultimately renders the notion of text ineffectual, since it looses any specificity and ceases to be capable of aiding in distinguishing different types of cultural artefacts.

Potentially more interesting than the question of the scope of text is what happens to the notion of text in a digital context. As digital textologists from Serge Lusignan (1985) and Richard Lanham (1993) to Espen Aarseth (1997) and Jerome McGann (2001) have observed, a fundamental epistemological shift occurs when moving from print to electronic textuality. In particular, although print text is composed of discreet symbols (characters) and is therefore, in a sense, already digital, the electronic medium is considerably better suited to infinite reorganizations and manipulations of those symbols; the computer makes such transformations trivial to accomplish. As a consequence, the electronic text is unstable: it is in a perpetual state of readiness to be reconfigured. And though a deformed electronic text may no longer be recognizable from its "original", it still retains associations with it through (undoable) algorithmic processes. Whereas print text can be thought of as a stable unit of meaningful characters, electronic text is better thought of as a dynamic process that encompasses several potential states for units of meaningful characters.

2. Analysis

Analysis is a classic 18th century practice worked out by John Locke and Etienne Condillac that has been adapted by humanities computing for a set of interpretative techniques that can be automated by the computer. Analysis stands in for various careful techniques of decomposing complex phenomena for the purposes of study. Digitization and computer-based tools provides us the ability to analyze large amounts of textual data quickly. This section will take three approaches to defining analysis in the context of humanities computing:

1. The difference between searching and analysis – We will look at how text analysis is different from everyday search features.
2. Five theses on analysis – We will present five theses on analysis in texual computing.
3. A reflection on analysis – We will walk through the process and results of a project to conduct analysis on analysis.

1. The difference between searching and analysis

One way into analysis is to look at what it is not. The tools of computer-assisted text analysis often resemble everyday tools. Word processors have searching tools that allow you to find a word or phrase. Such finding tools can be used as a simple text analysis environment. Likewise commercial search engines like Google do text analysis on a large scale over millions of web pages. Your word processor and Google are not, however, suited to searching large texts interactively, nor do they show you the results of a search in a way that can help you understand a literary text. Computer-assisted text analysis environments typically do three types of things beyond what the "Find" tool of a word processor might do:

i. Text analysis systems can search large texts quickly. They do this by preparing electronic indexes to the text so that the computer does not have to read sequentially through the entire text. When finding words can be done so quickly that it is "interactive", it changes how you can study the text - you can serendipitously explore without being frustrated by the slowness of the search process.

ii. Text analysis systems can conduct complex searches. Text analysis systems will often allow you to search for lists of words or for complex patterns of words, for example you can search for the cooccurence of two words or for the words before a pattern. Where you have structured text you can use the structure (typically TEI encoding) to ask questions about parts of the text.

iii. Text analysis systems can present the results in ways that suit the study of texts. Text analysis systems can display the results in a number of ways; for example, a Keyword In Context display shows you all the occurrences of the found word with one line of context as a concordance.

One can understand text analysis in the humanities as a convergence of traditions of interpretation in the humanities that evolved through print tools like the concordance with features of commercial text systems like rapid search and indexing. There is no simple history of text analysis. Instead there is a dialectic between the culture of computing and the culture of the humanities where both borrow ideas from the other. Visualization and text data mining are two new approaches that humanities computing is borrowing for analytical purposes. In the presentation we will briefly show some analytical tools borrowed from other traditions.

2. Five theses on analysis

Analysis is often understood as a set of techniques that involve the breaking apart of a complex into atomic parts for individual study. Whether it is a complex concept that is broken down into simpler concepts or a text broken into words (or characters), analysis starts with an interruption of a continuum into parts that can be synthesized into new representations. We propose these five theses on analysis as a way of analyzing analysis:

i. Analysis is not just about breaking down an object of study into parts. Every interruption of a continuous phenomenon like a text is also a synthesis – a building up of another representation. We don't access the atomic parts by themselves, they are always represented back to us in a new synthesis of parts that pretends to be atomic.

ii. Digitization is analysis. Analysis is usually thought of as a set of practices for the study of digital texts, but the choices made in the digitization of a text and its preparation for study constrain how the text can be broken apart by the computer. To give a simple example, a digital image of a document will have different atomic parts (pixels) that are amenable to analysis than a character string. Analysis starts with decisions about what to digitize, how to digitize the what, and what formats to use. The computer can only work with the data that was input. Garbage input, garbage analyzed.

iii. The analysis is in the interface. One form in which a text is represented is through the interface of our tools, including the tools of editing and research. The relatively low resolution computer tool forces texts to be broken into facets that you scroll through, page through or navigate with hypertext links. The design of reading interfaces can thus involve a breaking down imposed by the software.

iv. Text analysis is not neutral. The act of analysis changes the phenomenon analyzed. There is the illusion of stability – that we have texts that can be studied safely without affecting the original. We will argue that there is no original electronic text, only conditions of representation that change.

v. Text analysis is in a tradition of interpretation. What matters is the conversation we have through asking questions of others and other texts. Text analysis is one way to ask questions, but it is in a tradition that involves practices that are not automated. It is a moment of the humanities, one that may be gone.

3. A reflection on analysis

This section of the paper will close with a reflection on text analysis using text analysis. We will walk through a study on text analysis using tools available in the TAPoR portal. The analysis will be reflective in the sense that it will use text analysis on texts about text analysis.

i. We will show how one can build a corpus of materials about a concept like text analysis using portal tools like the Googlizer. Just-in-time tools that build on large search engines like Google provide a way of doing conceptual analysis on the fly. This will be compared with a prepared corpus like the abstracts for the ACH/ALLC 2005 at <web.uvic.ca/hrd/achallc2005/text_analysis.htm> . The abstracts database is available from the University of Victoria web site in XML and plain text.

ii. We will show how standard analytical tools that provide word frequency lists, collocates, and repeating patterns can help one think through how the phrase "text analysis" is used on the web.

iii. We will show how a simple visualization based on cluster analysis of the corpora can suggest anomalies for further thought.

3. Tools

What is a tool in the context of humanities computing? What would defining "tool" achieve? Like many of the concepts of humanities computing, those close at hand, like "tool", are often overlooked theoretically. Tools, as Heidegger reminds us, are things at hand that you pick up and use. Work is done through the tools, without reflecting on the tools, but on the interpretative work. A good tool disappears before the interpretative work. In scholarly work, however, there are moments when the assumptions encoded in tools and techniques need to be recovered, if only to ensure that the results of interpretative practices are consistent. In this paper we will look first at four relevant definitions of tool, and then how

Working Definitions of "Tool"

Here are four candidates for what a text analysis tool is that can be illustrated by the TAPoR portal:

1. Tool as Process. An automated process for the transformation of text data. In the case of humanities

computing the process would typically be for the transformation of linguistic data or strings and it would be a process that can be executed on a computer, but need not be. The earliest text analysis tools – concording tools – took tested and useful human processes and automated them.

TAPoR encodes this definition by distinguishing between texts and tools. The distinction seem uncontroversial, until one asks just what a text is, and especially what an electronic text it. In this paper we will illustrate some problem cases encountered in the design of TAPoR.

2. Tool as Program. A utility program that implements a process (see definition 1) that is packaged in a form that can be used easily on a computer. By this definition the program is the tool, not the process. Generally a tool is not a full-blown interactive application like a word processor.

By this account MS Word would not be a tool as you can use it interactively and you can use it to do many different things even if you could use it like a tool. Grep (global regular expression print), on the other hand, is a tool that does one task efficiently. Further, the UNIX notion of tools that can be piped together evolved in the extrication of utility processes from larger environments. (See Hauben and Hauben, Netizens, chapter 9)

TAPoR treats very specific things as tools. While there are lists of tools on the web, TAPoR priviledges web services that can be used through the portal. This has the advantage that one can try the tool, but it also limits the tools available and it presumes a model of what a tool is. A problem example, XTeXT, where "one" tool is represented in the portal as many tools, will be demonstrated.

3. Tool as Technique. An intellectual technique that involves transformative or interpretative practices defined with sufficient rigor that some of the practices might be automated on the computer as processes. A technique encompasses both the human and automated practices. Even more generally one can talk about methods that might be made up of various techniques.

One way to think of tools that goes back to Engelbart's work on augmentation and to think of a tool as something that extends our capacity to do intellectual work. The tool doesn't replace us, it extends our ability to accomplish tasks. What is important is the intellectual task and the techniques that can be adapted to the task. Within the context of a task a tool can automate some part of the technique used to achieve the task.

One of the weaknesses of a tool driven project like TAPoR is that it focuses on the tools not the techniques. The intellectual techniques are taught, trained, or played with, but they cannot be fully programmed. The human transformation of internalizing a technique to the point where tools can be used transparently needs support at this juncture in humanities computing too. Some of the extensions to the portal to support training will be demonstrated.

4. Tool as Environment. An interactive environment or game in which one can run a set of transformations for a single purpose. There is obviously a grey area between an atomic tool that does one thing (if we can imagine the doing of "one" thing) and an environment that serves multiple purposes. At what point does a tool get so much functionality that it becomes an environment for processes that isn't really ONE tool but more a workbench of tools? The point, however, is that we will call an environment a tool if it is used in a context for one end. Thus Excel becomes a tool if I just use it to sort columns of text.

This distinction between tool and environment is central to the design of the TAPoR portal. The portal is a particular type of environment (a communal portal) where one has access to tools (and other things.) TAPoR encodes this distinction, which in some cases, is a draw back. The artificiality of any interface paradigm can be seen when it breaks down. A good example in the case of TAPoR is the repositories of indexed texts. Are these tools or collections of texts? (In the presentation we will review a number of these anomalies.)

What can we learn from defining tools?

What is interesting about these definitions of tools is the reflection that goes into and through tools when they are designed and used. In the second part of this paper we will step back and look generally at the rhetoric of tools. In particular we will look at a history of the software tool as a primitive in Engelbart and in the development of UNIX. This sense of a tool, as in "grep is a tool", doesn't really get at whether processes, techniques and practices are tools at all, it maintains an analogy between a class of software and other practices. We can define what a tool is, but we have to ask if "tool" is the right thing to define in the first place. For many humanists, the word "tool" seems unsuited to humanities research. It smacks of the trades as if intellectual work was like joinery. If we look at Engelbart's language we see him using the woodworking tool analogy,

"A number of people, outside our research group here, maintain stoutly that a practical augmentation system should not require the human to have to do any computer programming--they feel that this is too specialized a capability to burden people with. Well, what that means in our eyes, if translated to a home workshop, would be like saying that you can't require the operating human to know how to adjust his tools, or set up jigs, or change drill sizes, and the like." (Engelbart, "Augmenting Human Intellect," section III.B.6)

The problem is the lack of alternatives to the tool analogy that can convey to humanists what utility programs can do. That said, we can imagine and will present an alternative analogy based on direct manipulation that would not represent text analysis as texts and tools, but as toys for manipulating texts in a game. This is the paradigm a study environment like the Ivanhoe game draws on. (See <www.speculativecomputing.org/ivanhoe/>)

An associated problem is the presumption that a tool is utilitarian - that is something used not for play, but for achieving a well defined goal, that it is a means not an end. Obviously for the tool designer the tool can be an end, but is it for user too? Users reflect on tools when they are learning them and when they break down. The experience of a new tool is not that of a known tool like a hammer, which one can pick up an use, unreflectively. A tool is not a tool at that moment of first encounter. It becomes a tool with repeated use or distraction. Humanities computing has a particular relationship with computing tools that can be seen by looking at a different discipline.

"Language is the principal - or perhaps the only - tool of the philosopher. For Wittgenstein, and for analytic philosophy in general, philosophy consists in clarifying how language can be used. The hope is that when language is used clearly, philosophical problems are found to dissolve." (Wikipedia, "Analytic philosophy" <en.wikipedia.org/wiki/Analytic_philosophy>)

Humanities computing also has a practice of clarification through tool use. Just as philosophy tries (and seems to repeatedly fail) to dissolve problems through careful language about language, humanities computing tries to engage problems through the development of computing tools, whether those tools are electronic editions, hypertexts, or text analysis programs. Encoding, in the sense of instantiating something in code, is itself a tool or practice that attempts to clarify the something sought. The problems we engage never dissolve; no tool answers our questions, that was a Wittgenstinian dream of a ladder that could be discarded. Rather, questions and problems tire and recede before new questions, like the philosophical question, what is a tool?

4. Text Analysis Tools

To this point, we have traced some of the evolution of the words "text," "analysis" and "tools" as they have transmuted over time through successive shifts in technology and practice. In contrast, the expression "text analysis tool" is a relatively recent composite term and much more closely tied to the specific contexts in which it is used (it cannot be examined generally, as we did with the other terms, because it is always already idiosyncratic for the circumstances in which it is used).

To conclude this essay, we will outline some of the ways in which the concept of text analysis tools has informed the development of the TAPoR Portal, but also how TAPoR has caused us to reconsider what we think of as text analysis tools.

References

Aarseth, E. J. (1997). Cybertext: Perspectives on Ergodic Literature. Baltimore and London: Johns Hopkins University Press.

de Condillac, E. B. (2001), Essay on the Origin of Human Knowledge, Trans. Aarsleff, H. Cambridge University Press, Cambridge.

Engelbart, D.C. (1962). "Augmenting Human Intellect: A Conceptual Framework," Summary Report AFOSR-3223 under Contract AF 49(638)-1024, SRI Project 3578 for Air Force Office of Scientific Research, Stanford Research Institute, Menlo Park, CA. Accessed online at <www.bootstrap.org/augdocs/

friedewald030402/augmentinghumanintellect/ahi62index.html>

Hanna, R (1998). "Conceptual analysis," In E. Craig (Ed.), Routledge Encyclopedia of Philosophy. London: Routledge. See <www.rep.routledge.com/article/U033>

Hauben, M. and R. Hauben (1997). Netizens: On the History and Impact of Usenet and the Internet, Wiley. A plain-text online version is available at <www.columbia.edu/~hauben/netbook/>.

Kernighan, B.W. and P.J. Plauger (1976). *Software Tools*, Addison-Wesley, Reading, MA.

Lancashire, I., J. Bradley, W. McCarty, M. Stairs, and T. R. Wooldridge (1996).*Using TACT with Electronic Texts*, The Modern Language Association of America: New York.

Lanham, R. A. (1993). *The Electronic Word: Democracy, Technology, and the Arts*. Chicago: University of Chicago Press.

Lusignan, S. (1985). Quelques réflexions sur le statut épistemologique du texte électronique. *Computers and the Humanities* 19, 209-12.

McGann, J. (2001). *Radiant Textuality : Literature after the World Wide Web*. Palgrave: New York.

JOINING UP THE DOTS: ISSUES IN INTERCONNECTING INDEPENDENT DIGITAL SCHOLARLY PROJECTS

Paul SPENCE
CCH, King's College London

John BRADLEY
King's College London

Paul VETCH
Centre for Computing in the Humanities, KCL

A characteristic of the Centre for Computing in the Humanities (CCH), King's College London, is its significant involvement with a large number of research projects that are producing digital products. At the present there are more than 30 in which our involvement is substantial, and for many of these projects our involvement stretches over a number of years.

During this time two related challenges have emerged. First, several of the projects naturally tend to group together – a user of one is likely to be interested in another as well. We have, at present, three significant groupings of this kind – a set of potentially interrelated projects about Anglo-Saxon England, a set of projects from the Classical period, and a set of Art History projects drawn from religious materials. Although the projects are done separately by different discipline specialists, there is some interest in sorting out ways that users can usefully switch from one to the other. The second challenge relates to the mix of technologies that each project uses. Some of these projects structure their materials in ways afforded by the relational model while others are using XML (primarily, of course, TEI). Eventually, perhaps, the tools available for XML will provide facilities comparable to the database engines available for the relational model, and at that point the XML model might well replace the relational one (see Bradley 2005 for some discussion about looking at XML in a relational sense), but at present this is not the case and we are finding that both our relationally-oriented projects and