

# Digital Studies / Le champ numérique, 2014

## The rise and fall tool-related topics in *CHum*

John Simpson, University of Alberta: [john.simpson@ualberta.ca](mailto:john.simpson@ualberta.ca)

Geoffrey Rockwell, University of Alberta: [geoffrey.rockwell@ualberta.ca](mailto:geoffrey.rockwell@ualberta.ca)

Amy Dyrbye, University of Alberta: [adyrbye@ualberta.ca](mailto:adyrbye@ualberta.ca)

Ryan Chartier, University of Alberta: [recharti@ualberta.ca](mailto:recharti@ualberta.ca)

Peer-reviewed by: Daniel Paul O'Donnell, University of Lethbridge; Michael Sinatra, Université de Montréal.

### Abstract / Résumé

What can we learn from the discourse around text tools? More than might be expected. The development of text analysis tools has been a feature of computing in the humanities since IBM supported Father Busa's production of the *Index Thomisticus* ([Tasman 1957](#)). Despite the importance of tools in the digital humanities (DH), few have looked at the discourse around tool development to understand how the research agenda changed over the years. Recognizing the need for such an investigation a corpus of articles from the entire run of *Computers and the Humanities* (*CHum*) was analyzed using both distant and close reading techniques. By analyzing this corpus using traditional category assignments alongside topic modelling and statistical analysis we are able to gain insight into how the digital humanities shaped itself and grew as a discipline in what can be considered its "middle years," from when the field professionalized (through the development of journals like *CHum*) to when it changed its name to "digital humanities." The initial results ([Simpson et al. 2013a](#); [Simpson et al. 2013b](#)), are at once informative and surprising, showing evidence of the maturation of the discipline and hinting at moments of change in editorial policy and the rise of the Internet as a new forum for delivering tools and information about them.

Quelles leçons pouvons-nous tirer du discours entourant les outils textuels? Davantage que ce que l'on pourrait croire. Le développement d'outils d'analyse de textes est une caractéristique informatique dans les sciences humaines depuis qu'IBM a commandité la production de *l'Index Thomisticus* ([Tasman 1957](#)) du père Roberto Busa. En dépit de l'importance des outils en sciences humaines numériques (DH), peu de personnes ont examiné le discours entourant le développement d'outils pour comprendre comment le programme de recherche a changé au cours des années. Reconnaissant le besoin d'effectuer ce genre d'enquête, un corpus d'articles provenant de l'ensemble complet de *Computers and the Humanities* (*CHum*) a été analysé au moyen de techniques de lecture à la fois éloignées et proches. En analysant ce corpus au moyen de répartitions de catégories traditionnelles ainsi que de modélisation de thèmes et d'analyse statistique, nous sommes en mesure d'acquérir des connaissances sur la façon dont les sciences numériques se sont façonnées et ont évolué comme discipline au cours de ce que l'on peut considérer comme ses « années intermédiaires », à partir du moment où le domaine est devenu professionnel (grâce au développement de publications comme *CHum*) jusqu'au moment où son nom a changé pour devenir « sciences humaines numériques ». Les premiers résultats ([Simpson et autres 2013a](#); [Simpson et autres 2013b](#)), sont à fois instructifs et surprenants, démontrant la preuve que la discipline est mature et faisant allusion à des moments de changements dans la politique éditoriale et à la montée de l'internet comme nouveau forum pour fournir des outils et de l'information à leur sujet.

### Keywords / Mots Clés

Computers and the humanities, *CHum*, tools, topic modelling, digital humanities history

### Contents / Contenu

- [Introduction](#)
- [Computers and the humanities](#)
- [Why tools?](#)
- [Methodology](#)
- [Outcomes and analysis](#)
- [Concluding with close reading](#)
- [Programming and tools](#)
- [Concordancing tools](#)
- [Works cited / Liste de références](#)

### Introduction

In what follows we describe *CHum* and its importance to the history and formation of the digital humanities, explain the focus on tool use, share the methodology of our analysis, and provide our results and interpretations after carrying out

the analysis in full.

## Computers and the humanities

Founded in 1966 by Professor Joseph Raben and printed by Queens College of the City University of New York, with financial support from IBM and US Steel, *Computers and the Humanities (CHum)* was the journal of the middle years of what we now know as the digital humanities. Over its run as the journal of the Association for Computing in the Humanities (ACH) from 1966 until 2004, after which it became *Language Resources and Evaluation*, *CHum* was one of the major venues for discussion, review and research by computing humanists. Surveying its run provides a summary representation of what were the important problems, questions, topics, methods, and tools through this period. This central role had much to do with the editorial vision that *CHum* embodied, leading prominent community member and *Humanist* editor Willard McCarty (2005) to remark:

Reading chronologically through *CHum* from the beginning is a salutary experience, especially for the first 25 years or so. One gains enormous admiration and respect for Raben's pioneering efforts, for the combination of energy, determination, deep insight and, above all, vision.

Joseph Raben served as editor of *CHum* from its inception in 1966 until 1986, volume 21, number 1. When he stepped down, the role of editor was passed to Glyn Holmes who was the associate editor. Holmes served as editor for the next ten years, stepping down in 1996 (v. 30, n. 1), when Daniel Greenstein took over the role and was then joined by Nancy Ide (v. 30, n. 2), which was overseen by guest editor Seth Katz. Shortly thereafter (v. 30, n. 3), Greenstein and Ide announced a refocusing/rejuvenation of the journal in an editorial. This partnership continued until Ide began editing alone in 1999 (v. 33, n. 3). The last editorial change, setting aside special issues overseen by guest editors, was when Elli Mylonas joined Ide in 2000 (v. 34, n. 1).

Despite editorial changes to *CHum* throughout its publication, the final years of publication showed a strong consistency with Raben's original vision of providing a cross-disciplinary venue focused on the use of computers in the humanities, describing itself in the final issue as follows:

*Computers and the Humanities* was established in 1966 to report on significant new research concerning the application of computer methods to humanities scholarship. Regular issues, special sections and special issues are devoted to reporting relevant computer-assisted work in a broad range of fields, including all the traditional humanities and arts related fields. In addition, *Chum* publishes work which presents theorized methodologies and applications relevant to the creation and use of digital texts and text corpora, hypertext, databases, images, sound, and multimedia. It also provides a forum for standards, resource management, and crucial legal and institutional concerns for the creation, management, and use of digital resources as well as their effective integration into scholarly teaching and research. ("Front matter" 2004)

With such a wide net of both content type and topics throughout its history *CHum* offers a unique and multifaceted lens that can be used to focus in on how the digital humanities took shape and grew. In this paper we will focus on one facet of the evolution of the field and that is the discourse around tools, specifically text analysis tools.

## Why tools?

Why should we be interested in tools in the digital humanities? Tools stand as physical manifestations of the methods used to satisfy the needs and desires of their users and so, with a little interpretation, tools may be used as proxies for understanding the agenda of their developers and users. In the case of digital humanists the tools in question stand at the intersection between theory and practice, acting at once as both means and ends. That we have more than just the tools themselves, having actual reviews about the tools and explanations of their development in *CHum*, makes considering them all the more valuable since the need to interpret the needs and desires of the creators and the users is significantly reduced. The availability of these direct commentaries makes this information significantly easier to obtain since many of these historical tools are notoriously difficult to run on modern machines, especially in ways that would resemble historical use cases.

Equally as important as any individual tool review, for the purpose of studying the evolution of the discourse around tools, is a critical mass of reviews and other tool related articles. *CHum* provides this quantity, being filled with tool reviews and opinion pieces about what tools should do. This begins with Louis Milic's opinion, titled "The next step," in the first article of the first issue, calling for more imaginative tool development: "Despite the odium which is likely to greet such an attempt, I should like to see the next step in literary computation to be truly imaginative." (Milic 1966, 5) This inaugural issue opened the doors to 500 contributions related to tool development and use to *CHum*, roughly 31% of the journal, over the 1609 total contributions made over next 38 years.

While it should be clear that *CHum* exhibits the extent to which tools were an important part of the digital humanities it does not seem to be the case that *CHum* was ever entirely clear about how to represent this. Yes, there are many articles that make important and essential references or descriptions of tools and methods using those tools, but there are only a few pure tool reviews over that time, 34, depending on what you count as a review. The first of these is seen in volume 10, number 1, 1976, when COCOA, a word count and concordance generator is reviewed (Day and Marriott 1976). This article is specifically called out as a "software review," receiving its own special section within the journal, though there had been reviews of programming languages, which could be considered a type of tool, and reviews of manuals for tools like the review of the *Manual for the printing of literary texts by computer* (Lieberman 1966) which was, in effect, a review of the PRORA tools. Perhaps the earliest example of an article focused on tools in the context of what one can do with a tool or tool methods was an article by Stanley Werbow (1967) in 1967 that reviewed Harold

Scholler's book *Word index to the "Nibelungenklage."* Werbow (1967) is understandably focused on the contribution of the book itself to the discipline, but in describing this he inevitably ends up describing and commenting on Scholler's method for producing the index.

Software reviews continue to be part of the journal on an intermittent basis until volume 24, issue 4, 1990, when they are replaced by "Technical reviews." This change in title was ostensibly made to remove potential confusion with the introduction of a section titled "Courseware reviews" in volume 23, number 6, 1989, since courseware is almost inevitably software too. This allowed the technical reviews to focus on research related tools (as is done in the research reported on here) and the courseware reviews to focus on tools more directly related to teaching and instruction.

It is interesting to note that *CHum* published hardware reviews as a separate section as well, but only three times, covering the Kurzweil Data Entry Machine (Galloway 1981), the AST Reach! modem with Crosstalk XVI (Pfaffenberger 1986), and the Hewlett and Packard Deskjet printer (Whitney 1989).

Technical reviews lasted almost ten years as their own section with the last one appearing in volume 31, number 3, 1997 with the title, "Dickens on disk" (Johnson 1997). Dedicated courseware reviews disappeared a bit earlier in volume 30, number 4, 1996/1997. From then on reviews were still published but they were typically written up within longer articles that included discussions of theory or methodology, such as "Multiple heuristics and their combination for automatic WordNet Mapping" (Lee, Lee, and Seo 2004), the last full article in the last published issue of *CHum*.

## Methodology

A four stage methodology was employed to investigate text analysis tool use as portrayed in *CHum* and then use this investigation to discuss the development of the digital humanities as a whole:

1. Gather a corpus by examining *CHum* articles by hand and extracting those focusing on text tools.
2. Automate the identification of additional articles mentioning tools.
3. Use topic modelling on the identified articles.
4. Return to the corpus of articles and read closely those that help understand the trends found through topic modelling.

Acquiring the initial corpus for examination was a relatively straightforward task that involved browsing the issues of *CHum* online and downloading the PDFs. This was carried out by a single researcher, taking approximately four months to complete in two separate passes. As odd as it may seem to have such a perfectly round number, this process revealed 500 articles with tools as an important focus, composed mostly of presentations of tools by their authors and reviews of tools by their users.

From this two-stage initial pass the names of all the text analysis or processing tools were collected and a search carried out across the entire corpus for all the occurrences of these tool names. This revealed the existence of an additional 167 articles that at least mentioned one of the previously identified tools, but which we did not include in further analysis since these were mentions and not the immediate focus.

This process of iterative expansions of selection naturally led to four classification categories for the articles that had been found: tool reviews (34), presentations of tools and reports on their development (42), descriptions of computer assisted research that may or may not have specifically mentioned a particular tool (400), and an "other" category (24). This last category consisted of articles ranging from a bibliography of articles to intelligent computer-assisted language learning to lists of tools.

With the tool related data available, making use of text mining methods became a fairly straightforward task. For this stage MALLET (McCallum 2002), a command line tool that implements the latent Dirichlet allocation (LDA) algorithm described by Blei et al. (2003), was used to perform topic modelling on the corpus of tool related articles with the issues of the journal grouped by year.

Topic modelling is a method of inferring what matters are dealt with across a corpus of texts by extracting words that appear together within documents into groups called topics. As suggested by Blei (2012), it is *as if* a reader used a series of coloured highlighters when reading through each text to colour each word related to a different topic a different colour and then collected the word of each colour together. Since the software tools that perform topic modelling do not know the meanings of the words they are grouping or have access to any information about how the words should be grouped other than how they appear in relation to each other across the collection of text and within each of the documents the best that can be done is to approximate this behaviour.

What the topic modelling tools do is attempt to construct a conditional distribution of topics across the entire corpus from which the corpus could have been generated via random selection, each document in the corpus is made up of all the topics but in different proportions, and a conditional distribution of words within each topic, each topic is made up of all the words from the corpus but in different proportions. The final outcome of this process is a set of topics with associated weights or strengths for the corpus as a whole, each with a list of all the words (less stop words that are seen to have little meaning such as "the," "to," and "a") in the corpus that have associated weights or strengths within each topic. Topic modelling is thus a process of producing collections of words from a corpus of documents that, given that collection of documents, seem to be related.

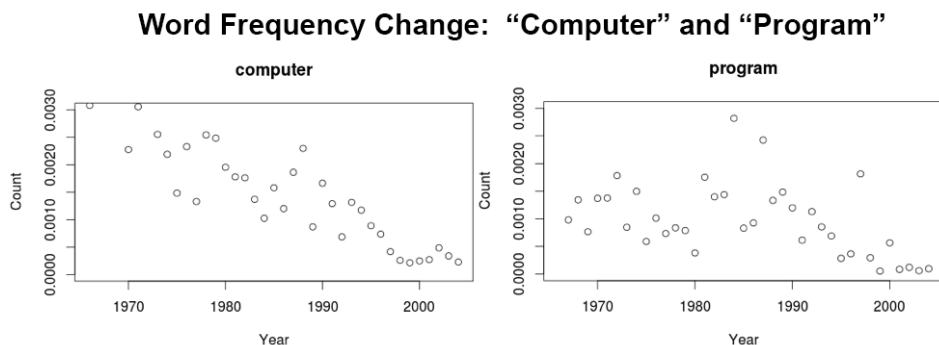
It is important to remember that since the topics are really just words that the topic modelling algorithm has determined are probably related that the topics themselves come without meaningful names and often receive a name simply for referential purposes such as "Topic 48." Any further name other than a list of the words in the topic in decreasing order of strength is the construct of some outside interpretative process, typically a human judgement, and may or may not adequately represent the topic.

With the topics and related frequencies provided as data from MALLET the statistical programming language R was used to batch process the output, producing a series of graphs displaying the rise and fall of the topics identified by MALLET as existing in *CHum* from 1966 to 2004.

## Outcomes and analysis

One of the first outcomes of interest that we discovered was a steady decrease in the frequency of the word "computer" over the course of run of the journal, while "program" remained relatively constant until the very final years of the journal, when it all but flatlines (Figure 1). The decline in these two words is also seen in *Humanist* (Figure 2), an electronic discussion forum that has grown into a major communications hub for the digital humanities since its creation in May 1985.

Figure 1: Single words with a high prominence or degree of change.

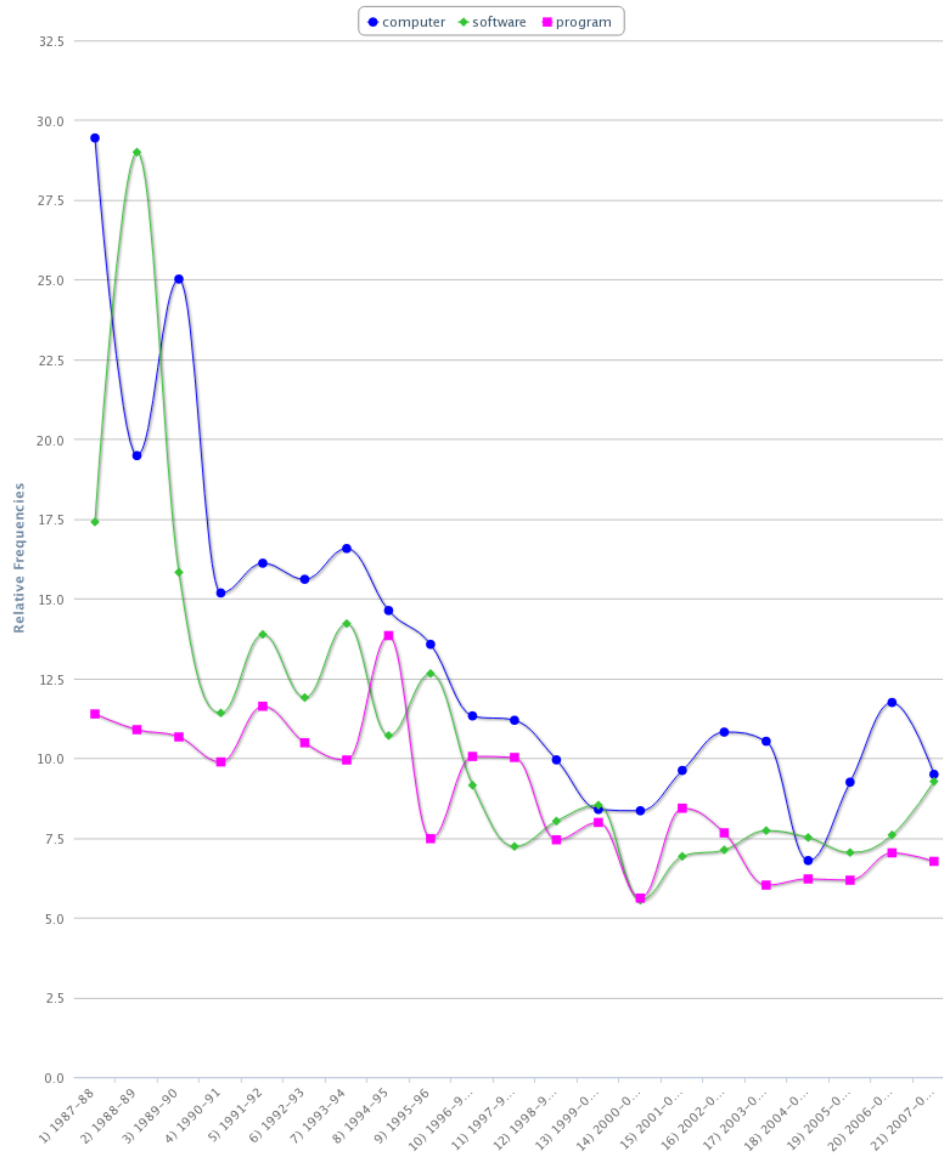


The changing distribution of these two words tells of the transition from humanities computing to digital humanities through a shift in emphasis from hardware to more ephemeral digital services. The rise of the Internet and the Web in the early 1990s are contributing factors to this shift. The site of the computational work changes as researchers begin to develop and use new information sharing network services like discussion lists (e.g. *Humanist*) and then web-based tools like TACTweb. As the Web rises in importance, attention shifts from what any particular computer can do with the programs installed directly on it to what can be accomplished by computers more generally, as instruments of digital media.

Shifts in the words used throughout these reviews are also indicative of the maturation of the digital humanities as it grew from a discipline that used computers and co-opted the language of computation and the associated methodologies from other disciplines into a discipline with its own vision about framing and pursuing research questions.

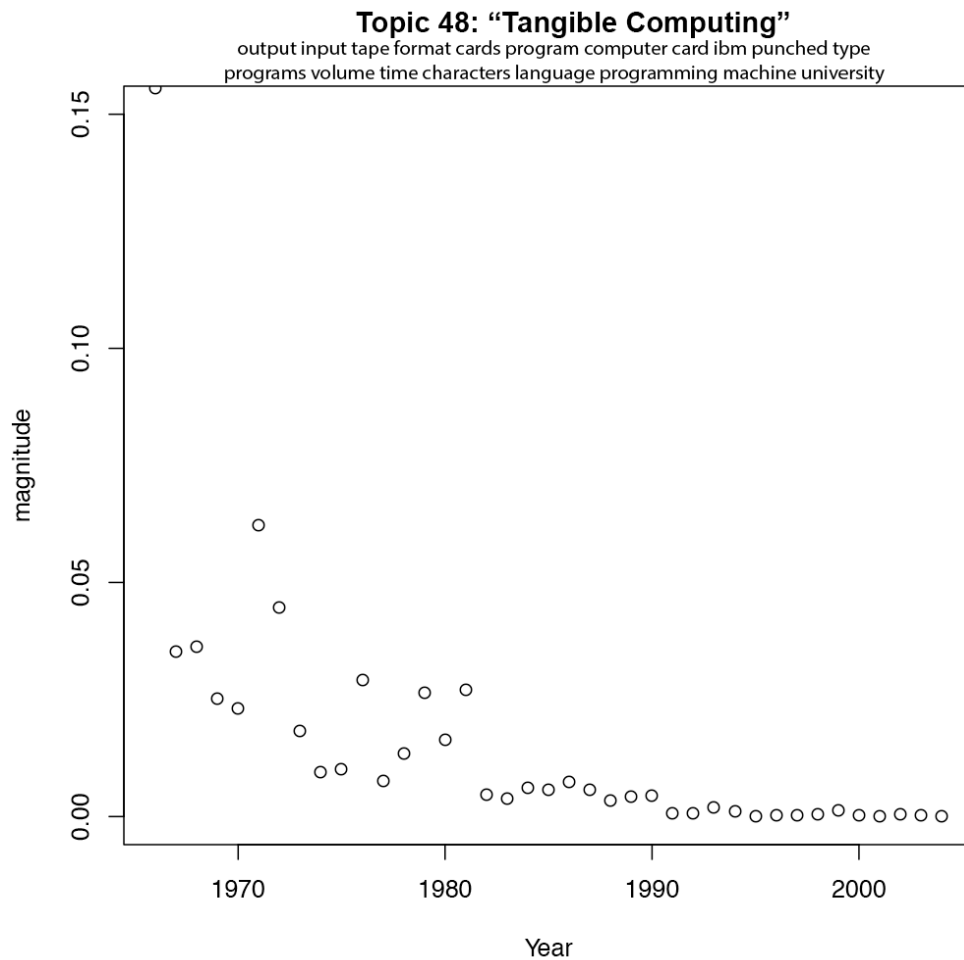
None of these shifts happen suddenly or completely, much processing still takes place on desktop processors and the influence of computer science methodologies remains strong in the digital humanities, but they do happen and the resulting change in how we think and talk about the digital humanities is observable in the results of the analysis carried out here.

Figure 2: The decline in "computer", "software", and "program" on *Humanist*.



The suggestion that there is a shift in the thinking about digital humanities that goes from tangible hardware to intangible concepts such as data and information on the basis of a few changing word frequencies seems to be a bit of a speculative stretch. Indeed, this would be exactly what it would remain if not backed up by additional effort and research. Traditionally, this additional work would come in the form of a careful reading of the texts in question, followed by a nuanced argument of the observations made, possibly attending to declared shifts in editorial direction or other meta-level information related to the journal but standing slightly outside its core content, the articles themselves. Instead of this sort of work we performed topic modelling on the *CHum* data using MALLET, producing groups of words with relatively high frequencies of occurrence in close proximity and then recording their relative frequency of occurrence across the entire corpus for each year.

Figure 3: Topic 48, a strong decrease in Tangible Computing.

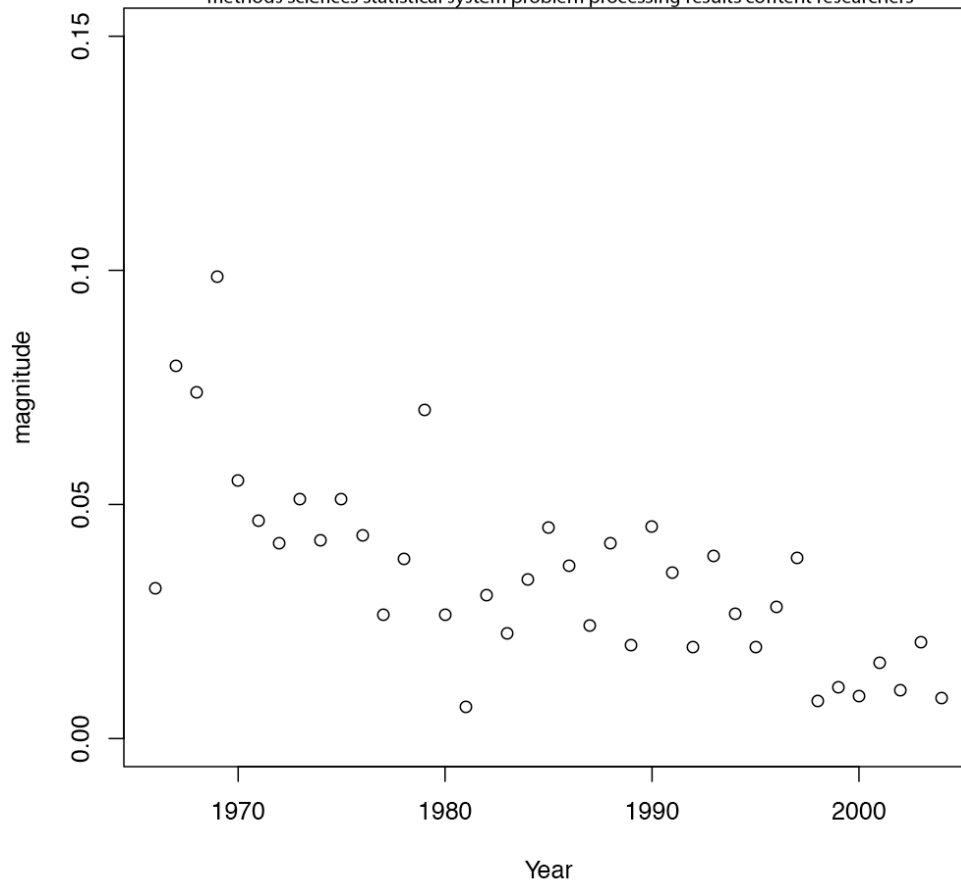


Topic 48 (Figure 3: "Tangible Computing") includes words like "tape," "punched," "machine," and "programming" which stand together as a clear reference to the sorts of batch programming with punch cards on a computer mainframe that was still being used as late as the early 1980s. While such computing might be described as "physical computing" all computing is a physical process (Deutsch 1997) and so describing it as "tangible" computing is more apt.

Figure 4: Topic 42, decrease in Scientific Method.

### Topic 42: "Scientific Method"

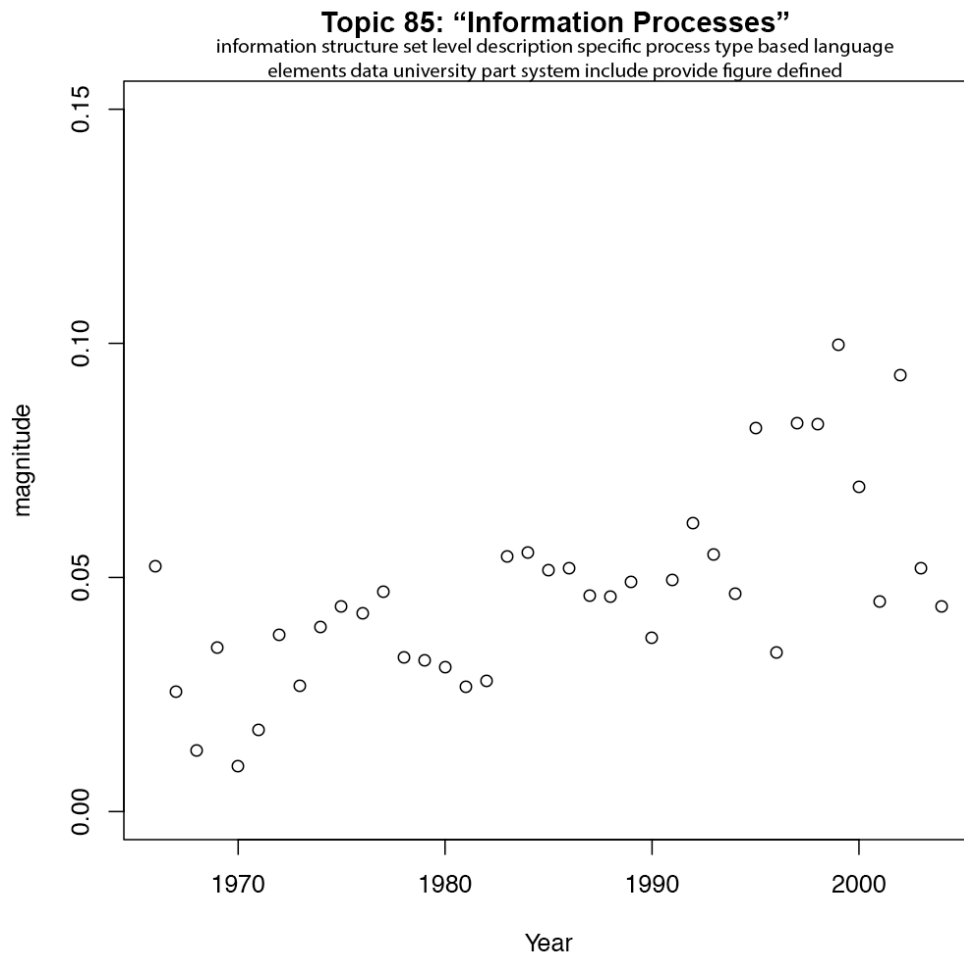
computer analysis research computers data study humanities problems science studies  
methods sciences statistical system problem processing results content researchers



Topic 42 (Figure 4: "Scientific Method") is still focused on the material conditions of computing but with the inclusion of procedures and methodologies directing the use of such things. With words like "sciences," "analysis," "statistical," and "results" this topic indicates the extent to which scientific methods were influential in directing the growth of the digital humanities. Understood in this way the chart makes clear that initially the particularly quantitative methods of the sciences held a larger influence over tool development and assessment in the late sixties and early seventies than they would continue to have in the early two-thousands. This would fit with the popularity of stylistics or the quantitative/statistical study of style popular in the field at that time.

In contrast to the decline seen in topics 42 and 48, topic 85 (Figure 5: "Information Processes") is rising and doing so and at roughly the rate by which the two topics just discussed were declining. Featuring words like "information," "process," "data," and "figure," topic 85 also suggests that the means by which information is being described, displayed, and ultimately justified is changing. Understood in contrast with the changes shown in topics 42 and 48, 85 shows an embracing of tools that provide the ability to use a language of proof and justification of the processes embedded in these tools based on the underlying information and the display of the results. In this way it shows the rising interest in the intangible aspects of the digital humanities.

*Figure 5: Topic 85, increase in Information Processes.*



## Concluding with close reading

We now turn from the distant reading of topic modelling to a close reading of the discourse around tools in articles in *CHum*. Our distant reading suggested that there was a shift in the discourse as digital humanists moved from tools on mainframes to microcomputer programs in the early 1980s and another shift reflecting the move to web services in the mid 1990s. Topic 48 (Figure 3) associates with mainframe tools with words like "cards," "IBM," "punched," and "tape." Words like "computer" and "program" last until the mid 1990s after which digital humanists shift to focusing on the web and web resources/services. To understand these transitions in a more nuanced way we returned to the early articles, lists, and reviews about tools, reading selected papers more closely.

## Programming and tools

The first point to make about the discourse around tools in the early issues of *CHum* is that it is as much about programming as tools. In the late 1960s when *CHum* got started there were no tools in the sense of packaged software you could buy or download. Tools were programs developed in specific programming languages that ran on specific machines. You corresponded with someone distributing the code and asked to be sent a copy of the program on punch cards or tape. It would take a certain technical skill and familiarity with the programming language to get the program running. One can see this in the list of "Computer programs designed to solve humanistic problems" published in the second issue of *CHum* (1966). This list is made of items like this one for a tool called *Discon*:

Discon. Purpose of program: Concordance-making. This program is simply the well-known DISCON, originally written for the 7090 and converted to the 7040 by R. L. Priore of Roswell Park Memorial Institute, Buffalo, New York, then converted to the 7044 here by Marjorie Schultz. Type and format of input: Punched cards: 6 cols. ID, the rest data.

Programming language used: Fortran IV. Required hardware: IBM 7044. Running time: Approx. 4 min. per 1000 lines of poetry, exclusive of printout time.

Correspond with Roger Clark or Lewis Sawin, 123 W Hellems, University of Colorado, Boulder, Colorado. (p. 39)

Later on there would be debates about whether computing humanists should learn to program or just use applications, but in the late 1960s programming languages were thought of as tools (with which to build higher-order tools). In early issues of *CHum* the question was whether you learned to program or depended on a programmer, and if you learned,



what language should you learn. Louis Milic (1966), in the opinion mentioned above that opened the first issue of the journal, "The next step," set the tone by arguing that "As long as the scholar is dependent on the programmer, he will be held to projects which do not begin to take account of the real complexity and the potential beauty of the instrument." (p. 4) He returns to this topic in a later an essay, "Winged words" (Milic 1967), where he concludes,

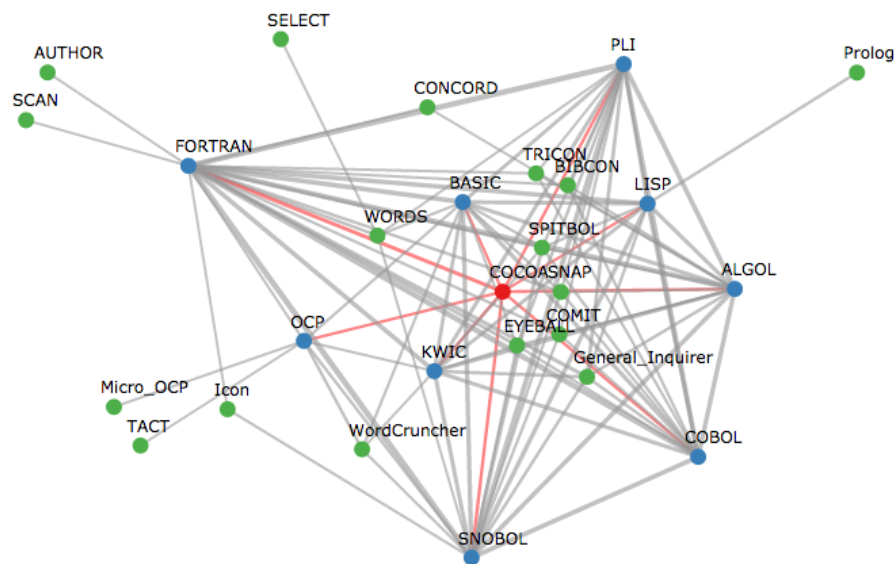
the scholar must learn to be his own programmer. That is axiomatic. He cannot learn what the computer can do if he has to ask another to interpret for him. With the development of new high-level languages like SNOBOL, competence in which can be acquired even by the stiff reflexes of the middle-aged scholar, though not without effort, there can be no excuse for remaining technologically illiterate.

It is no surprise therefore that for decades there would be reviews of programming languages suitable for humanists. For example, in the second issue of *CHum* there was a review of PL/I by Heller and Logemann titled "PL/I: A programming language for humanities research" (1966) which outlined what humanists needed and why PL/I would suit:

Humanities programmers are concerned primarily with symbolic data as opposed to numbers. They wish to manipulate sentences and texts, or strings of their own notations representing a symphony, the structure of a painting, the movements in a drama, or a collection of artifacts they have uncovered. We observe that PL/I allows data to take the form of strings of symbols. (p. 21)

Later there were articles about SNOBOL (1967 v. 1, n. 5), SNAP (1970 v. 4, n. 4), more PL/1 (1972 v. 6, n. 5), Icon (1982 v. 21, n. 1), and ProIcon (1990 v. 24, n. 4). There were also articles on teaching humanities computing that mentioned languages like BASIC as they became popular teaching languages. There were reviews of books that included code or taught humanists how to program in languages like Pascal. It is beyond the scope of this paper to follow the evolution of how programming languages were discussed in *CHum*, the point here is that programming languages were tied to tools in the early years, as is illustrated in Figure 6, "Tool Network," a social network graph illustrating tools and programming languages mentioned together within the papers of our corpus.

Figure 6: Tool network with COCOA selected (<http://i.artsrn.ualberta.ca/viz/about.php?var=network>).



## Concordancing tools

The second and final point to be made about the tools discussed is the importance of concordancing tools. The very first issue of *CHum* carried a review of the *Manual for the printing of literary texts by computer* a manual for the PRORA series of concordancing tools developed by Robert Jay Glickman and Gerrit Joseph Staalman at the University of Toronto. The list of "Computer programs designed to solve humanistic problems" in the second issue includes a number of concordancing tools including *A Concordance Generator*, *Discon*, *Drexel Concordance Program*, *Concordance*, *BIGCON*, and *PRORA* (multiple programs). Louis Milic in "Winged words" (1967) recognized that of the projects listed a previous issue, the largest class was concordancing or indexing projects:

There are 120 projects listed under "Literature," though some scholars are responsible for more than one. Under examination, these break down into the following components. Predictably enough, the largest class (53) consists of concordances, dictionaries, word-lists, indexes, and catalogues of lexical items. (p. 28)

What were these concordancing tools and how did they evolve? Drawing on articles and reviews we can reconstitute a history (among many) of key text tools that evolved from mainframe tools for generating concordances to interactive tools suitable doing text analysis directly:

**COCOA:** Developed initially at the Atlas Computing Laboratory in the 1960s, COCOA (COunt and CONcording on the Atlas) was one of the first concordance tools adapted to be machine independent. It was distributed on some 4000 punch cards. COCOA influenced OCP and its text tagging system was used for decades by tools like TACT. (Day and Marriott 1976)

**OCP (Oxford Concordance Program):** Developed by Ian Marriot and Susan Hockey in 1979-1980, OCP was a pivotal tool in that it was "the first machine-independent, reasonably priced, user-friendly program for humanists interested in computers applications in natural language but unwilling to learn a programming language in the process." (O'Brien 1986, 141)

**Micro-OCP:** As the title suggests, *Micro-OCP* was a version of OCP (Version 2) for the IBM PC compatible microcomputer. Developed in the late 1980s it was, like OCP, a batch program, with a menu driven interface and a command language. (Jones 1989) It was soon superseded by interactive analytical environments.

**WordCruncher and TACT (Text Analysis Computing Tools):** WordCruncher and TACT were the first widely available interactive text analysis programs. WordCruncher, developed in the late 1980s and sold by the Electronic Text Corporation, was based on the Brigham Young University Concordance Program. (Olsen 1987) TACT was designed by John Bradley and developed by a team at the University of Toronto in the late 1980s. It was released in 1989 as freeware for the IBM PC and has features still not found in web tools. Being free and interactive made it easier to using in humanities computing courses. (Hawthorne 1994)

It should be mentioned that there were many other concordance tools discussed in *CHum* and important tools like ARRAS that didn't get reviewed in *CHum*. This history is specific to the journal, but it does touch on most of the influential tools that were available for use (and review.) It should also be mentioned that there were other classes of text analysis software that got sustained attention in *CHum* including stylistic analysis tools like EYEBALL and content analysis tools the Harvard General Inquirer, but those are for another article.

To conclude, the creation of tools, especially text tools, has been constitutive of the digital humanities, at least in the middle years. The founding of the digital humanities is often said to have begun with Father Roberto Busa approaching IBM founder Thomas Watson in 1949 to ask for the use of a computer that would ultimately lead him to produce the *Index Thomisticus* (Busa 1980). While this may have been the founding moment of the discipline it was not until two decades later that a critical mass of people had gathered through conferences and scholarly communications tools like journals. *Computers and the Humanities*, which ran from 1966 until 2004, was arguably *the* journal of the field we now call the digital humanities and it certainly played a formative role along with other journals like *Literary and Linguistic Computing* and discussion lists like Humanist. Reading the reviews and articles published in *CHum* over its 38 year run therefore provides a lens with which to focus on how the digital humanities shaped itself and grew as a discipline in the middle years. In this article we have used this lens to show how the discourse around text tools evolved from language tied to mainframe text processing to language around information services. This lens also shows us the early importance of text tools starting with concordance/indexing tools and the programming languages they were developed in.

---

## Works Cited / Liste de références

Blei, David M. 2012. "Probabilistic topic models." *Communications of the ACM* 55.4: 77–84.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. "Latent dirichlet allocation." *The Journal of Machine Learning Research* 3: 993–1022.

Busa, R. 1980. "The annals of humanities computing: The Index Thomisticus." *Computers and the Humanities*. 14.2: 83-90.

"Computer programs designed to solve humanistic problems." 1966. *Computers and the Humanities*. 1.2: 39-55. <http://www.jstor.org/stable/30199206>

Day, C, and I. Marriott. 1976. "Review of COCOA: A word count and concordance generator." *Computers and the Humanities*. 10.1: 56. <http://www.jstor.org/stable/30204221>

Deutsch, David. 1997. *The fabric of reality: The science of parallel universes and its implications*. United States of America: Allen Lane/The Penguin Press.

"Front matter." 2004. *Computers and the Humanities* 38.4: I-ii. <http://www.jstor.org/stable/30204949>

Galloway, Patricia. 1981. "The Kurzweil data entry machine." *Computers and the Humanities* 15.3: 183–85. <http://www.jstor.org/stable/30199974>

Hawthorne, M. 1994. "The computer in literary analysis: Using TACT with students." *Computers and the Humanities* 28.1: 19-27. <http://www.jstor.org/stable/30200307>

Heller, J. and G. W. Logemann. 1966. "PL/I: A programming language for humanities research." *Computers and the Humanities* 1.2: 19-27. <http://www.jstor.org/stable/30199201>

Johnson, Eric. 1997. "Dickens on disk." *Computers and the Humanities* 31.3: 257–60. <http://www.jstor.org/stable/30200426>

Jones, R. L. 1989. "Micro-OCP." *Computers and the Humanities* 23.2: 131-135. <http://www.jstor.org/stable/30200151>

- Lee, Changki, Gary Geunbae Lee, and Jungyun Seo. 2004. "Multiple heuristics and their combination for automatic WordNet mapping." *Computers and the Humanities*. 38.4: 437-55. <http://www.istor.org/stable/30204955>
- Lieberman, David. 1966. "Manual for the printing of literary texts by computer by Robert Jay Glickman and Gerrit Joseph Staalman." *Computers and the Humanities*. 1.1: 12. <http://www.istor.org/stable/30199195>
- McCallum, Andrew Kachites. 2002. "MALLET: A machine learning for language toolkit." <http://mallet.cs.umass.edu>.
- McCarty, Willard. March 5, 2005. "18.615 computers and the humanities 1966-2004." *Humanist Discussion Group*. <http://dhhumanist.org/Archives/Virginia/v18/0604.html>.
- Milic, Louis. 1966. "The next step." *Computers and the Humanities* 1.1: 3-6. <http://www.istor.org/stable/30200056>
- Milic, Louis. 1967. "Winged words: Varieties of computer application to literature." *Computers and the Humanities*. 2.1: 24-31. <http://www.istor.org/stable/30203947>
- O'Brien, F. 1986. "Oxford concordance program." *Computers and the Humanities* 20.2: 138-141. <http://www.istor.org/stable/30200056>
- Olsen, Mark. 1987. "Textbase for humanities applications: WordCruncher." *Computers and the Humanities*. 21.4: 255-260. <http://www.istor.org/stable/30207396>
- Pfaffenberger, Bryan. 1986. "AST reach! Modem and crosstalk XVI." *Computers and the Humanities*. 20.2: 147-48. <http://www.istor.org/stable/30200059>
- Simpson, J., G. Rockwell, S. Sinclair, A. Dyrbye, R. Chartier, M. Radzikowska, and R. Wilson. 2013b. "Just what do they do? On the use of text analysis in the humanities." Paper presented at *CSDH/SCHN 2013*, University of Victoria, Victoria, British Columbia, Canada.
- Simpson, J., G. Rockwell, S. Sinclair, K. Uszkalo, S. Brown, A. Dyrbye, R. Chartier. 2013a. "Framework for testing text analysis and mining tools." Poster presented at the *Digital Humanities 2013*, University of Nebraska-Lincoln, Lincoln, Nebraska, USA.
- Tasman, P. 1957. "Literary data processing." *IBM Journal of Research and Development*. 1.3: 249-256.
- Werbow, Stanley N. 1967. "The first computer-assisted MHG Word-Index." *Computers and the Humanities*. 2.1: 51-53. <http://www.istor.org/stable/30203954>

