Scholarship in interaction: case studies at the intersection of codework and textual scholarship

Zundert, J.J. van

Summary

Hermeneutics, or the theory and method of interpretation, has a long history that is deeply intertwined with the philosophical and exegetic roots of the type of textual scholarship that developed in Europe. Postmodern relativism and the transient, shapeshifting nature of digital technology are disconcerting forces for a textual scholarship in this tradition that is concerned with human culture memory function and that is therefore fact and stability oriented. However, stability is romantic. Any textual scholarship work must succumb to what could be called the "hermeneutic condition" – there exists nothing but interpretation.

When moving towards post-digital forms of interpretation, understanding the relation between hermeneutics, textual scholarship, and digital technology is an urgent and important academic need. An intimate understanding of digital code and how it affects textual scholarship is needed, but no structural scientific agenda to do so exists. In part this is the result of a lingering uneasiness between those scholars with a vested interest in digital (or "quantitative") approaches and those on the "qualitative" side of the spectrum. This methodological anxiety became more profound in the last two decades of the twentieth century and inhibited a sincere discourse concerning digital hermeneutics to develop within the digital humanities (DH). More recently some inroads seem to develop (cf. work by e.g. Rockwell, Stephen Ramsay, and Rafael Capurro). However, in debates there is still a tendency to reduce the potential of computation to a methodology of quantification and reduction (with, for instance, Johanna Drucker as a strong proponent of this argument). DH has a nature that is more hybrid than these debates suggest and there is not an a priori discontinuity with the hermeneutic traditions. Even if many computational methods in DH today lean towards quantified and reductive techniques, formalizations and analysis patterns

are not hermeneutics-free. Just as philological practice fundamentally cannot escape hermeneutics, neither modeling nor quantification can escape the hermeneutics involved in choosing the assumptions on which their formalizations are based, nor can they escape the hermeneutics involved with the interpretation of their results. Thus, as Katherine Hayles states, the tension between algorithmic analysis and hermeneutic close reading should not be overstated. However, a real challenge is, as inter alia David Berry and Federica Frabetti point out, that we need to understand the internal hermeneutics of code and computational methods to their core. We can only gain this understanding from a deep intimate engagement with "digitality" and software code itself.

Particular effects of the interaction between digital technology and textual scholarship do little to further this engagement. These effects may be uncovered using science and technology studies methods that offer a framework to systematically observe and reflect on scientific processes. In doing so, it appears that, for instance, a graphical user interface suggests a fictitious transparency of model and paradigm. The graphical interface is as much an opaque barrier to the internal paradigm of a system as it is a means of engaging with that very system. The interface is not the model, but a visualization that adds its own limitations, interpretations, and affordances. If interfaces are mostly formulated as expressions of an existing paradigm, they may well inhibit the appreciation and adoption of the affordances of new models and paradigms. This is for instance the case in textual scholarship where most digital editions are essentially digital remediations of books. They are based on a fully mimetic representational philosophy in which the digital edition is nothing more or less than a visualization that mimics as closely as possible the physical print document, which negates the affordances of other possible models. This effect I call paradigmatic regression, and it provides one explanation why extremely little experimentation with alternative modeling possibilities are found in textual scholarship. Such experimentation could even be based on very elementary and widely available Web technology. HTTP-links could be used to model intertextuality, for instance, which is a potential that has been pointed out well also within the textual scholarship community by, for example, George Landow and Theodor Nelson. Of course this paradigmatic effect is also linked to the conscience theoretical and practical

choices that textual scholars make, both individually and as an academic community. The strong document-mimetic tenets of TEI-XML and the inviability of, for instance, the "social edition" suggest that these choices are in a certain sense conservative and in favor of a paradigm in which the ownership of the edition is still fully appropriated by the scholarly editor and in which the model is dominated by the format and limitations of the codex.

There is little evidence that the interaction between textual scholarship and digital technology results in significant methodological change. An expected effect of such change would be the development of a methodological pidgin and the exchange of technical and methodological vocabulary. The case studies presented in this thesis – mostly in the context of research at the Huygens Institute – do not provide examples of such exchange beyond some project management terminology that does not strictly relate to textual scholarship methods or computational analysis. What change there is seems to be introduced rather "covertly" at the level of programmed models and statistical analysis, the technical details of which are again covered over by graphical interfaces.

Paradigmatic regression effect is a divergent force that drives close and distant reading "schools" further apart rather than closer to each other. Scholarly editors and computational literary researchers supposedly cater to each other's needs. However, digital documentary editions are mostly computationally inaccessible monolithic XML-tag "cathedrals", while computational literary research projects produce mostly unsustainable "raw" text streams and annotations. A better common model is needed if the strengths of both need to apply to textual scholarship and digital scholarly editing. Knowledge graphs could provide such a model.

Arguably textual scholarship should take an interest in models of description and analysis that are more adequately geared towards digital objects. First of all because there is a general trend towards softwarization in society, resulting in ever more cultural objects and processes being expressed as digital objects and processes. Secondly because a similar softwarization process, more covertly than not, affects textual scholarship methods. If scholarly tasks and decisions are delegated to software engineers, and if many of these decisions and tasks are delegated by the programmers to the code that they

develop, then it becomes utmost important that textual scholarship develops and adopts the means to make such delegated processes and decisions scientifically accountable in the case of bespoke code that is applied in scholarship. Contrary to what pernicious metaphors suggest, software is not a neutral and objective tool. Software code is a product of human creativity that harbors inbuilt assumptions about and models of reality. Thus, to warrant scientific process and quality control, a sufficient framework for code peer review and the critical study of code should be in place, so that engineers that create bespoke scholarly software can be held to scientific accountability and responsibility. The boundary between code that should be submitted to a form of code peer review and code that need not, is not exactly defined. Obviously the inner workings of a statistical computer language like R do not require scholarly peer review, but just the understanding that it has been rigorously examined and tested by computer scientists and statisticians. But code purposefully written – even if using, for instance, R – to execute a specific scholarly task in the context of a one-time-use research design, does require peer review. Notwithstanding the admirable achievements of software studies, critical code studies, and media-studies with regard to critical interrogating the role of software in society, such a framework for peer review of bespoke code applied in scholarship does not exist. Code peer review however, requires code literacy. And code literacy is a rare commodity in academic scholarship.

When code replaces certain tasks in scholarship this warrants questioning in some detail the role of those in scholarship that write the code. In a sense programmers are becoming scholarly authors and editors. Thus a discussion on authorship and editorship is relevant. Very concisely put the intellectual and philosophical history of authorship is a discourse on the question who "owns" the authority to determine the meaning of the text. Depending on time and context the answer to this question has varied between "deities" or "God" (in more historical times), "the author", and "the reader" (in more recent times). The claims to this authority over interpretation are related to claims on who gets to decide about truth. Obviously post-structuralist relativism and the "death of the author" recast any such claims as subjective.

Scholarly editing by and large responded in two ways to the post-structuralist "crisis" of factuality, determinability, and authority. One was by proposing

archive-like and open forms of scholarly editions that fully recognize the intersubjectivity of editing and interpretation. This direction in scholarly editing recognizes the fluidity of text and the process-like nature of authoring, editing, and reading. It favors inclusivity of materials and readership. Ideally all possible resources related to a text should be included in an edition – which might rather be called "archive" in that case. The process of editing should be as open as possible and the edition itself should be open ended, so readers are able to add their own interpretations (e.g. by adding notes and annotations). Web 2.0 and digital technology in general are considered ideal means to realize this type of editions and editorial processes. The other response is a strong retreat on documentary editing. Rather than exploring new models, this direction in scholarly editing aims to impose the existing model of the book on digital technology. It reasserts and reaffirms the primacy of "philological fact" over interpretation. It is not uncommon for practitioners and theorists in textual scholarship to sway between these extremes, as for instance Jerome McGann did.

A closer inspection of the roles of author, editor, and engineer, reveals that writing code is a form of authorship. Both editing and programming in the context of textual scholarship turn out to be forms of revisionary authorship. However, in the majority of cases coding in a scholarly context is a form of unclaimed or, worse, misappropriated authorship. The authorship of code inserts at least one layer of interpretation into the process of scholarly editing in comparison to traditional modes of editing. This creates two problems. Firstly, the text and its interpretation potentially become even more fluid and unstable. This is contrary to the aim of philology and scholarly editing, which can generally be understood as an attempt to stabilize the text. Of course this notion of stabilizing the text is highly problematic in itself, but adding the authoring and performance of digital code into the process of editing is likely to further destabilize the text and its interpretation. One reason for this is that code is endowed with "deferred agency": its execution may result in autonomous scholarly decisions. Another reason is that code also has its own rhetoric and its own performativity, that are both still poorly understood. The second problem is that all possible effects of coding as a form of revisionary authorship within textual scholarship go unchecked by current scholarly processes. The reason for this is that programming is not

recognized as authoring nor as scholarly editing, neither by scholars nor programmers. Therefore the potential influence of programming on scholarly editing, irrespective of its aims, goes largely unevaluated.

The introduction of the programmer – or computer science expert – into the scholarly process, raises an as of yet unanswered question of accountability. If digital objects and code are an integral part of the scholarly argument then their producers have an obligation to claim the contribution they make to the argument. Not just to make a righteous claim to academic credit, but foremost in order to be accountable for the scholarly argument they co-create. The intent of the revisionary authorship that code authoring produces, is not self-evident and not self-explanatory. This is true for any form of authorship. As early as the 1980s Donald Knuth acknowledged that code has rhetorical functions and that code authoring has poetics – if mostly only tacitly. He provided a model and implementation of a system to express the rhetoric and poetics of code more explicitly. Unfortunately his rationale has since been poorly understood and taught, and has been applied even less. Such rationales deserve revisiting and programmers should claim explicitly the revisionary authorship they appropriate through executable code.

The case studies in this dissertation show that an interdisciplinary community of developers and textual scholars is actively seeking new points of contact between scholarship and digital models of text. Different models foreground different aspects. TEI-XML seems in the majority of its applications to foreground the model of the codex and the structures of text in printed books. Statistical models in distant reading analyses foreground the idea of text as meaningfully quantifiable strings of tokens. Both models produce highly sophisticated and valuable interpretations. However, both models also put forward a particularly narrow understanding of texts as either predominantly document structures or "bags of words". They are demonstrable "lossy" with regard to the objects they describe, which is easily to gauge from the fact that they both forgo most material aspects of any text. This is not to say these models are fundamentally unable to capture and express these aspects, but rather to say that their particular makeup introduces a selective focus on the aspects they capture particularly well. If you use a "lens" that is exceptionally good at describing text structure, text structure is analytically foregrounded. If you use a lens that excels at describing bags of words,

all texts start to look like bags of words. If you only have a hammer, many things look like nails.

The traditional analytical means of textual scholarship, to which for instance the classical apparatus belongs, are models too. Any model imparts some specific effect to interpretation. New models create new affordances that may lead to new perspectives and that may uncover new knowledge. As argued however, textual scholarship exhibits paradigmatic regression that inhibits productive experimentation with new models. The overwhelming freedom of modeling that the digital environment offers may be in part an explanation for this regression. That freedom stands in stark contrast to the specific constraints of epistemological devices such as an "apparatus criticus", that have been developed over multiple centuries and have been shaped to certain extents by the particular materiality of the codex. But in the interest of exploring new epistemologies for textual scholarship it would serve to also liberate texts from such known models and to examine the affordances of new and different models.

Case studies point in the direction of graphs as powerful means to express the multidimensionality of text. This suggests that the graph model is an interesting candidate to be one of these possible new models to be applied in textual scholarship. It can be succinctly shown that graph models offer a number of benefits over traditional models, such as feasibly solving TEI-XML overlap, representing multidimensionality, and computationally "mapping" textual variation.

Reflection on methodology is one of the most important obligations within any scientific domain. The social sciences, especially in the shape of science and technology studies, provide effective tools and techniques for such reflection, for instance through ethnography and autoethnography. Many more case studies of the interaction between computer science, software engineering, and the humanities will be needed before any firm conclusion can be drawn about what constitutes viable digital technological innovation in the latter, strongly hermeneutic domain. An autoethnography and case studies of attempts at such innovation in the textual scholarship domain at the Huygens Institute show that it is very difficult to innovate methodology while appreciating and safeguarding hermeneutic approach. It appeared that both

scholars and technologists should strive to be much more aware of their own methodological limitations, and that both groups should accept the scientific strengths of the other group as complementing the weaknesses of their own group. In this respect, an autoenthnography of fifteen years experience of working on the intersection of these domains shows that both scholars and technologists need to be more respectfully aware of each others' scientific qualities, and that they need to be more aware of their individual and shared scientific responsibilities in methodological innovation, accountability, and reflection.

Current computational approaches to hermeneutic scholarship are too much rooted in the rhetoric of speed and scale of automation, driving attention away of a particular hard but interesting computational problem to explore: that of computational hermeneutics. Scholars should strive to be both less intimidated by, and more welcoming to computational explorations. While doing so they should be confident and fully aware of the immeasurable scientific wealth and value incurred by more than twenty centuries of hermeneutics.