Lost in Pools of Data

Boot, P.

published in
New Technologies and Renaissance Studies III
2022

document version
Version created as part of publication process; publisher's layout; not normally made publicly available

Lost in Pools of Data: Text Reuse in the Emblem Genre and the Nature of Humanities Research Data

Peter Boot
Department of Literary Studies,
Huygens Institute for the History of the Netherlands
peter.boot@huygens.knaw.nl

Introduction

This essay will be about a limitation of the tools that we develop in the digital humanities. Its central thesis is that the tools we design to answer our questions need to be embedded in an environment that helps us organize, evaluate, and annotate their outcomes. Tool outcomes provide not answers but data that need interpretation (Sculley and Pasanek 2008). Each outcome is the result of a number of choices. The interpretation of the tool's outcomes requires an awareness of the choices that were made in its application. And as the researcher usually applies his or her tools multiple times, each time with different choices for settings and input material, each time with (slightly or radically) different outcomes, the question for the researcher begins to move. What was a question about, say, the topics in a collection of medieval texts, turns into a question of keeping track of input and output of multiple tool runs, motivations for choosing certain settings, judgments about the usefulness of certain outputs, etc. The core intellectual problem that the tool was designed to solve gives way to a more general information overload problem that was out of scope in its development—or more probably not even considered as an issue.

This is not just a concern for the individual researcher trying to make sense of the data. It has a wider impact when considering the transparency of the research process and the need to avoid cherry-picking our results. It is relevant when we want to share our data with other researchers. And it becomes particularly urgent when we want to use our tools for distant reading applications.

These general questions will be discussed in the context of quotation and reference in the Renaissance genre of the emblem. The widely popular emblem (in its simple form, a motto, an engraving or woodcut, and an explanatory poem) reused sayings, concepts, and pictorial motifs from older literature

ISBN 978-1-64959-017-6 (pdf) ISBN 978-1-64959-016-9 (paper) ISBN 978-1-64959-037-4 (epub)
New Technologies and Renaissance Studies III (2022) 33–62
(New Technologies in Medieval and Renaissance Studies 9)
https://www.itergateway.org/resources/new-technologies-and-renaissance-studies-III

(Bible, classical authors, wisdom literature), and in its turn its contents were reused in later writings and visual culture. The detection of text reuse (a younger text reusing shorter or longer pieces of an older text) looks like a simple problem that the computer should be able to handle. And indeed, there are many digital humanities projects that have developed more or less sophisticated tools for finding quotations, allusions, or overlap between texts more generally.

To begin, I will discuss our experiences in developing and using text reuse detection software in the context of the Nederlab digital library. Nederlab is an ambitious digital library project that brings together for research purposes the main Dutch collections of digitized texts. In a pilot project, designed to test the usability of the Nederlab collections, we developed software to locate text reuse in emblem texts or reuse of emblem texts in later literature. This turned out not to be an easy task, but what is most relevant in the context of this essay is that the work isn't done when the parallel texts, i.e., potential quotations, have been identified. The real work is then only beginning. What is a relevant parallel? When two texts use the same Bible quotation, this is text reuse, but the younger text is probably not quoting the older. How do we compare the output of multiple tool runs with different settings and slightly modified input files? If we have decided a certain parallel is irrelevant, how do we avoid the same parallel turning up again in subsequent runs?

In another experiment, I wrote a prototype for a management tool that provided support for the information management needs that intensive use of a text reuse tool brings with it. The text reuse detection tool compared two corpora (e.g., the books of the Vulgate Bible and a collection of emblem books) in search of text parallels. The management tool stored all input parameters, hashes of input files, and all output for each run of the text detection tool, and made it possible to explore and annotate the information by run, by corpus, or by individual file. Discussion of this management tool will show how such a tool can indeed provide some overview of the text reuse detection runs and their results, though some aspects remain elusive.

However, the question about this meta-level tool that I will discuss is what the need for such a tool shows us about the nature of humanities data and the fundamental characteristics of tool use and computation in the humanities. For many of the choices a researcher has to make, be they in text reuse detection, in authorship attribution, in topic modeling, or really anywhere, there are no *a priori* reasonable settings. Always when we compute, the details of how we compute affect the outcome. As there is usually no obviously correct

way to compute, we are faced with a choice between multiple outcomes. For our own overview we need a facility to compare and annotate multiple outcomes, but even without taking into account our own needs, we would have a scholarly obligation to store the outcomes of our tool runs and to give account of the choices that we made.

In conclusion, I will look at the question of what this means for the tools that we develop. Facilities like the management tool discussed here should help humanities researchers, in our case Renaissance scholars and medievalists, manage the information overload resulting from using powerful and complicated tools, and focus on the core investigation. They should help other researchers evaluate and replicate scholarship. But does this imply that all tools should be embedded into a virtual research environment for organizing and evaluating their output? Could such a facility be sharable? Maybe tools could provide hooks for extension with facilities for further analysis? It is in no way certain that general-purpose management tools are actually feasible; at the level of the individual application, they may be prohibitively expensive. At present, there does not seem to be an easy way out of this dilemma.

Tools for detecting text reuse

Scholars have always been interested in text reuse, be it in the form of allusion, quotation, or unacknowledged borrowing of text. Text reuse can indicate literary influence, and a recognized quotation can clarify an otherwise obscure passage. Especially in ages that didn't value originality as much as we do, borrowing text was also an accepted way of building literary or philosophical works. As text reuse seems a more or less objective, simple phenomenon that just requires patience to find, its detection seems an ideal task for the computer (at least in those cases where the text is quoted in its original language). A number of digital humanities projects have developed software to do so. Mark Olsen and colleagues worked on detecting borrowed text in the *Encyclopédie* (Olsen, Horton, and Roe 2011), similar work on medieval encyclopaedic texts was done at Monash University (Zahora, Nikulin, Mews, and Squire 2015). Kane and Tompa developed a text detection tool, for a collection of Latin quotations, to help find both the sources of the quotations and later reuse (Kane and Tompa 2011). The *Tesserae* project is specifically interested in finding allusions, rather than full quotations (Coffee et al. 2012). There is a long list of related work, some of it, like the work at Monash, inspired by plagiarism detection software.

Text reuse in the emblem

One of the genres where text reuse detection seems particularly appropriate is the emblem book. The emblem was created by Italian lawyer Andrea Alciato, who published his *Emblematum liber* in 1531.The emblem as he created it would develop over the next two centuries from a learned game for humanists to a vehicle for mass moralization, but the essential ingredients remained the same: image and text (motto and subscription) jointly providing a witty, edifying, or instructive message. The emblem, originally in Latin, spread all over Europe, and emblem books in the various vernaculars soon began to appear.

From the beginning, emblem book writers took pride in reusing older texts and images. The sources that were used included medieval bestiaries, classical coins and medals, ideas about Egyptian hieroglyphs, as well as classical history, mythology, and literature (Daly 1998; Moseley 1989). For explicitly Christian emblem books, the Bible also became an important source. Henkel and Schöne's *Emblemata* (1976) shows the extent to which emblem writers reused the same pictorial motifs. Bath (1994) stresses how much the emblem book owed to learned collections of proverbs and sayings, such as Erasmus's *Adagia*, and notes that emblem books in their turn also contributed to these commonplace books, such as Mirabellius's *Polyanthea*. The emblem's influence on later literature was discussed by Daly (1998). The pictorial motifs used in emblem books were often reused in paintings (De Jongh 2000), in churches (Cieslak 1995) and as "applied emblems" in, for example, interior decoration (Heckscher and Wirth 1959).

In this essay, I will look specifically at text reuse in or based on emblems from the Low Countries (Porteman 1977; Stronks 2008). The Low Countries produced their fair share of Neolatin emblem books, as well as multilingual books and books in Dutch. The two main topics for Dutch emblem books were love and religion, often combined. The love emblem, though not strictly a Dutch invention (Saunders 2007) was brought into blossom in the Netherlands in the early seventeenth century in a series of successful emblem books, such as Otto van Veen's *Amorum emblemata* (Van Veen 1607/8). Van Veen's book, featuring Cupid on every page, was simultaneously published in multiple editions, each containing texts in a different combination of languages. The love emblem shared the emblem's predilection for intertextuality (Bloemendal 2007), reusing, for example, the vocabulary of Petrarchism and quoting canonical classical authors. It was turned to religious use by Van Veen himself (Van Veen 1615) in his *Amoris Divini Emblemata* (Boot 2012). This

book reused texts and motifs from the books about secular love in writing about religious love, but replaced the classical authors by the Bible and the church fathers. Jacob Cats wrote realistic emblems about love, but added a social and religious application (Cats 1618). A series of other books, such as Van Leuven's *Amoris divini et humani antipathia* (Van Leuven 1629), opposed secular and religious love (with a preference for the religious variety, of course). All these books, and others in the same genre, borrowed heavily from each other, with respect to emblem concept, texts and pictorial motifs. The most prominent later Dutch emblematist was Jan Luyken. After *Duytse Lier* (Dutch lyre), an emblematically embellished book of poems and songs about love (Luyken 1671) and an emblem book titled *Jesus en de ziel* (Jesus and the soul) (Luyken 1685), he published a series of religious emblem books with emblems based on everyday objects and activities, such as *Het leerzaam huisraad* (Instructive household objects) (Luyken 1711). In his later emblem books, each emblem is followed by one or two pages of Bible quotations. I used Luyken's books to test systematically the best approaches to detect as many quotations as possible. Another book that I will look at is Willem den Elger's *Zinne-beelden der liefde* (Emblems of love) (Den Elger 1703). Den Elger reuses images from the religious emblem books' tradition in a mostly secular setting. The reason his book is interesting is that he quotes many other poets and emblem writers, and it is therefore suitable material for evaluating techniques for text reuse detection.

In this essay, the focus is on how the introduction of digital tools often does not solve the problem that they were introduced to solve. They do contribute to a solution, of course, but they also move the problem to a new level of abstraction which may at a cognitive level be harder to manage than the original problem. Before moving to this (meta-)issue, let me summarize what we found with respect to text reuse in the (Dutch) emblem genre. We discovered hardly any intellectual debts that were previously unknown. The forms of text reuse that we encountered were primarily shared quotations. Emblem book authors do not usually quote longer text fragments from each other. What they quote are usually other quotations. (From each other, they quote mottoes, but these are so short they escape detection; they are often only two or three words long.) The classical authors and the Bible are the main sources for these quotations. The favorite classical author is Ovid, which is unsurprising given the Dutch emblem book's focus on love. Another form of text reuse occurs in those emblem books that mix the emblem and the song book genre, such as the *Nieuwen ieucht spieghel* (New mirror of youth) (Anonymous 1617). There we encounter many songs from earlier collections of songs, such as *Den nieuwen verbeterden lust-hof* (The new improved garden

of delights) (Vlack 1607). We encountered hardly any reuse of emblem texts in later works, with the exception of explicitly anthological works such as *Een lees- en zangboekjen voor de jeugd* (A reading and song book for young people; 1853).

Detecting text reuse in Nederlab

Nederlab (http://www.nederlab.nl/) is a digital library targeted at researchers. It brings together the most important collections of digital text from the Netherlands: the collections of newspapers, journals, and books digitized by the National Library, the collections from the *Digital Library of Dutch Literature* (*DBNL*), now also managed by the National Library, as well as linguistic corpora and other texts; collections from other institutions will follow shortly. To test the usability of the Nederlab infrastructure, in 2015 we looked into the possibility of doing text reuse detection on the Nederlab text collections.[1] In our research we looked into the sources for emblem text as well as the reuse of emblem texts in later texts. We created four collections of text: emblem books available within Nederlab (39 titles), Dutch Bible translations (6 titles), poetry until 1750 (199 titles) and books for children and young people (466 titles). We started looking for textual parallels without specific hypotheses about the quantity or type of text reuse that we were going to detect or about the books that we were going to find it in.

A simple text reuse detection tool was developed, inspired by the Text Pair software developed for the *Encyclopédie* project (Olsen and Horton 2009). The logic of that program was rewritten from Perl into Ruby, because Ruby was a better fit for our infrastructure. The idea was that after the experiment, the resulting software could be implemented within the Nederlab environment for wider use. It is fair to say that we underestimated the effort required to replicate the Text Pair functionality, and what we could deliver in the limited amount of time available for the project was only a limited amount of functionality. The software compares a query text file against an indexed base corpus, looks for corresponding shingles (n-grams of words), and merges adjoining hits into larger parallels. Some options are available to fine-tune the creation of the n-grams: n itself is a parameter and could be set to 3, 4, 5, or any other value; words from a stopword list could be ignored; a minimum word length could be imposed; words could be cut off after a certain length; and some elementary rewrite rules could be applied

[1] This was an "in kind" contribution to Nederlab development financed by the Huygens Institute for the History of the Netherlands. The researcher on the project was the author of this essay; development work was done by Meindert Kroese.

to combat orthographical variation.[2] As output, the tool produced a CSV file with pointers to the parallel texts (based on filename and position in the XML hierarchy) and the parallel text fragments with some context. Table 1 shows a brief extract of such a CSV file. The file in the first column is the States' Bible; the files in the third column are various emblem books.[3]

File location 1	Parallel in file 1	File location 2	Parallel in file 2
_sta001stat01_01.TEI.2.(…).p.17100	Uytspansel in * het midden der Wateren; * ende dat make	brun001embl02_01.TEI.2.(…).p.1456	bevrijd waren, en door * het midden der wateren, * dwers door die dorre woestijne
_sta001stat01_01.TEI.2.(...).p.17191	maeckte * die twee groote Lichten: dat groote licht tot heerschappye * des	luyk001schr02_01.TEI.2.(...).p.978	maakte * die twee groote Lichten: dat groote Licht tot heerschappye * des daags
_sta001stat01_01.TEI.2.(...).p.17191	kleyne * Licht tot heerschappye des nachts; * oock de Sterren.	luyk001schr02_01.TEI.2.(...).p.978	groote * Licht tot heerschappye des * daags,
_sta001stat01_01.TEI.2.(...).p.17191	kleyne * Licht tot heerschappye des nachts; * oock de Sterren.	luyk001schr02_01.TEI.2.(...).p.978	kleine * Licht tot heerschappye des nachts; * ook de Sterren.
_sta001stat01_01.TEI.2.(...).p.17203	te heerschen in * den dach, ende in de nacht, * ende om scheydinge	hard001godd02_01.TEI.2.(…).q.24463	verheventheyt Godts. Dat ick * den dach ende nacht in dese * mijne pelgrimagie

Table 1: Extract of CSV file giving potential parallels. The words delimited by asterisks are the parallel, the rest is context (here shortened). The references to the file locations are also shortened. The extract shows both true and false parallels.

Unfortunately, what transpired immediately is that there is no single best setting for the parameters: no setting, certainly, that is applicable for all research questions and all corpora, but even for a single corpus it is necessary to run this tool, or probably any tool, a number of times with different parameter settings. That necessity arises mostly because quotations are almost never literal, character-by-character, identical to the text they quote (Kolak and Schilit 2008). When the Bible says "Honor thy father and thy mother," the text may be quoted as "We were told to honor our fathers and our mothers," and these texts only share the words "honor" and "and." For historic texts, an additional complication is that spelling was much more variable than today.

The parameters serve to limit the impact of this (usually irrelevant) variation. Ignoring words from a stopword list, usually of frequently occurring

[2] We hardly used the rewrite facility. Creating a set of acceptable rewrite rules would have been a project in itself.

[3] The parallels all supposedly refer back to verses in the first chapter of Genesis. The first parallel, however ("het midden der wateren" [in the midst of the waters]), comes from a retelling of the story of Exodus in De Brune's *Emblemata of Zinne-werck* (De Brune 1636). The next three are true quotations, the last one is again a coincidental agreement in word usage.

function words, would in the example probably remove "and," "thy," and "our," reducing the relevant texts to "Honor father mother" and "honor fathers mothers." Ignoring words below a minimum word length can have a similar effect. Cutting off words beyond a certain length (say six) would make that "Honor father mother" and "honor father mother." Apart from the capital, the texts are now identical and would be selected by our algorithm—that is, if we look for trigrams. If we were looking for four-grams, we would find nothing, as in the present context there are only three words left. So even a literal quotation of "Honor thy father and thy mother" would be overlooked if we applied stopword filtering with four-grams. This illustrates that parameter settings can work both ways: they can remove irrelevant variation, increasing the possibility of finding a quotation, but they can also remove evidence for quotation, and thus decrease that possibility.

Another way in which a parameter setting can be a double-edged sword is by increasing the number of located parallel texts while at the same time increasing the number of false hits. When we cut off words after a certain length, or replace words by their lemmas, we lessen the amount of variation and therefore increase the number of hits, true or false. Technically, we may increase recall (the proportion of true parallels retrieved by our system) at the cost of diminishing precision (the proportion of true parallels among our hits). This is especially visible when switching for instance from four-grams to trigrams. By loosening the requirements, the number of hits increases spectacularly. In a run where we compared the Dutch States Bible, the authoritative 1637 Bible translation (Anonymous 1637), against the Nederlab emblem corpus, 6,866 potential parallels were found when looking for four-grams, but no fewer than 49,836 when looking for trigrams (other settings being equal: minimum word size three, word cut-off length six, simple spelling uniformization, and no stopword removal). But often these potential parallels are the results of very generic trigrams (in Dutch) such as "ende dat ghy" (and that you/thou) or "den houwelicken staet" (the married state). On the other hand, the trigrams run does locate a number of true parallels that the four-grams run does not, such as the words from Genesis that God "schiep den Mensche" (created man). Inconsistent spelling is by far the most important reason why trigrams retrieve parallels that cannot be located using four-grams: Luyken's *Schriftuurlyke geschiedenissen en gelykenissen* (Scriptural histories and parables) (Luyken 1712) actually contains the entire Bible verse (Gen 1:27), as well as a wider context, but the words before and after "schiep den Mensche" are spelled differently by Luyken. It is clear that with 49,836 potential parallels, it has become impossible to check them manually.[4]

[4] For some projects, this problem of low precision becomes so urgent that the main

I give a few more examples, all from the book of Genesis, illustrating how a single run, with one choice of parameters, is insufficient. Using five-grams and minimum word length three, we find the parallel "Licht tot heerschappye des nachts" (light to rule the night) between the States Bible and Luyken; the parallel disappears when we use four-grams and minimum word length four (other settings being equal). Conversely, the second setting detects "mensche alleen zy; ick sal hem eene hulpe" ([It is not good that the] man should be alone; I will make him a helper). Both combinations of settings also report false parallels that the other one does not locate (e.g., "des eenen tegen den anderen" [of the one for the other], again a very general formulation from II Macc. 14), and there are true parallels that neither locates, as when Luyken quotes Gen 1:14. Neither run is by itself sufficient, nor is their combination. Similar examples could be given for many other combinations of parameter settings.

Once we have found the textual parallels, the work of evaluation begins. Is the text parallel really an example of the younger text quoting the older? In some cases, clearly not: for example, the parallel text may occur in an editorial note from a later age. In other cases, the parallel text may be from the printing privilege. In many cases, what we find are shared quotations: both the emblem book and a later volume of poetry quoting an older text—for our corpus, mostly the Bible and classical authors. This is no doubt a significant form of text reuse and shows an important fact about the textual culture of the time, but it does not show that the author of the younger work quoted, or even knew, the older work.

If each run of the text reuse detection tool is bound to retrieve a number of true and a number of false parallels, where the true ones may or may not be relevant to the research question, and if it is necessary to run the tool multiple times with different parameter settings, this implies that the researcher will need to inspect a large number of CSV files. Many of the detected parallels will be identical, some will overlap, some will occur only once. The question of the researcher now changes from "is there meaningful text reuse in these text collections?" to "what parameter settings are suitable for this corpus?" (based on language, quality of transcription, spelling variation, quotation density, and size), or "how do I find the new parallels in this CSV file given the ones I have already inspected?" or "what parameter settings did I use for this run again?" The researcher needs to keep track of

---

task switches from locating (potential) parallels to filtering out unlikely parallels (Forstall, Coffee, Buck, Roache, and Jacobson 2015).

the encountered true and false parallels as well as the settings they resulted from and will need some form of automated support for that.

Managing the text reuse research data

This management problem was the starting point for a second experiment. Here I decided to use the Text Pair software *as is*. The goal was not to develop software that would become part of a maintainable infrastructure but to use a suitable tool for detecting textual parallels; to build onto that tool an experimental environment for displaying and annotating parallels, and thus to evaluate the multiple parameter settings that the tool could use. To what extent could such a tool overcome the information overload problem?

I identified five main requirements for this management tool:

1. Store data. It should store all the settings that were used in the index and query steps of the detection process and it should store the information about the retrieved parallels.

2. Display. Based on the stored information it should allow display of a detection run, the parameters that were used, and the parallels that were found. It should also be able to display together the parallels that were found between the same text pair in different runs, and to display retrieved parallels based on a query on the parameter settings (for instance display the textual parallels found between text a and text b when using stopword removal but no minimum word length). This requirement is the only one that is often, to some extent, implemented in text reuse detection tools.

3. Annotate. It should facilitate annotation at all levels of the reuse detection process—run, parameter setting, text collection, text, and parallel—and on combinations of those levels, such as the usefulness of a parameter setting for a certain pair of texts in run x. While for most annotations free-format text is probably sufficient, for the retrieved parallels it should be possible to state unambiguously whether they should be considered true or false positives. This requirement relates to the next ones, about summary statistics and auto-annotation.

4. Compute summary statistics. When the researcher has annotated the retrieved parallels for a certain detection run as true or false positives, the tool should be able to compute for that run the standard information retrieval quality measures: recall, precision, and F1. See below for further discussion of the applicability of these measures in this context.

5. Auto-annotate. One of the biggest issues for the researcher in evaluating multiple parameter settings is having to find new parallels in a run among the many parallels he or she has already inspected. The management tool should therefore be able to auto-annotate parallels as true or false based on annotations to parallels from earlier runs.

With respect to the summary statistics: they should deliver a first and rough estimate of the quality of a run using a certain set of parameters. It is important to understand that, as we have no ground truth for the true amount of quotation in our corpora, the quality of a run can only be judged in terms of the amount of quotation that it detects with respect to other runs of the tool. We define recall as the overlap between the parallels detected in a run and the true (approved) parallels from all runs on the same corpus pair. Formally, for a run r, the recall for that run is defined as:

$$Rr = \text{length}(APar \cap Par_r) \,/\, \text{length}(APar)$$

where APar is the set of approved parallels for the corpus pair and Parr is the set of parallels detected in run r. Similarly, the precision is defined as the approved parallels in a run as a fraction of all parallels retrieved in that run, technically:

$$Pr = \text{length}\,(APar \cap Par_r) \,/\, \text{length}((APar \cup RejPar) \cap Parr)$$

where RejPar is the set of rejected parallels between the two corpora. Both precision and recall vary between 0 and 1, and are ideally 1. $F1_r$ the overall quality measure, is the harmonic mean between P_r and R_r. For each of the three numbers, we also define a count-based variant, where we don't take into account the length of the parallels, but just their numbers.[5] The values of these statistics change, and approach their true values, as more detection runs are done (or more precisely, as the results of more runs are annotated as either relevant or not). It remains up to the researcher to choose a criterion for approving or rejecting parallels: Is textual agreement in itself sufficient to approve a parallel? Or should there be more than mere linguistic agreement and should the texts actually quote a same text, or should one text quote the other? The latter can never be decided based on local textual evidence alone, but requires as a minimum knowledge about the dates of origin of the texts as well as knowledge about the textual traditions in which they were written.

[5] Definitions inspired by Potthast, Stein, Barrón-Cedeño, and Rosso 2010.

I developed a tool that fulfilled most of these requirements.[6] The text::pair index and query script were modified to so as to save the input parameters and the output into a MongoDB database. An application was developed to browse the database, to annotate its contents, and to compute the summary statistics.[7]

The next figures show some runs where I compared the Den Elger book with a merged Nederlab collection, including other emblem books, books for younger people, poetry, and bibles. The spelling of all texts has been uniformized based on a series of regular expression.[8] Figure 1 shows an overview of runs. For each run, we see some information about the corpora that were compared to each other and the summary statistics, if they were computed. From here it is possible to view all parameter settings for the run, to view the retrieved parallels at file pair level, or (using the filter options) to view the information at file pair level for multiple runs, selecting by query corpus, index corpus, or both. Figure 2 shows a selection of results at the corpus pair level, sorted by query file (text b). We see that parallels were retrieved for the text with filename _aem001aem01_01.txt (*Een Aemstelredams amoureus lietboeck* [An Amsterdam book of love songs]) (Anonymous 1589) in runs 251 and 259. These parallels (1 in run 251, 3 in 259) were rejected (that's what the "n" in "3n" says). We also see the total length for these parallels. The "notes" option gives access to an annotation window, the "view" option to the retrieved hits and the "filter" options allows display of hits filtered by indexed (base) text, query text or both. Figure 3 shows the hits for De Brune's *Emblemata of Zinne-werck* (De Brune 1636) in run 255. The one in red has been rejected ("het krieken van den dag" is a current expression meaning "at the crack of dawn"), the two in green have been accepted. They are quotations from Ovid's *Remedia Amoris* and Horace's *Epistles*. The "y" (yes) and "n" (no) buttons are used to approve or reject the quotations. At the bottom of the screen we see a view of both texts with the hyperlinked quotations in red. Figure 4, finally, shows the hits in different runs between the same two texts, here sorted by their offset in the first text. It allows us to inspect which parallels are detected in which runs. We see one hit ("amor formae condimentum," as mangled by the spelling uniformization) that was only detected once, in run 259.

---

[6] It doesn't do auto-annotation (requirement no. 5) or retrieval by parameter setting (part of no. 2).

[7] Developed in Python, using MongoDB, the Bottle Web framework and the jQuery Datatables plugin.

[8] Uniformized, not modernized or regularized. The purpose was to map spelling variants to the same form, not to create a modern or correct spelling.

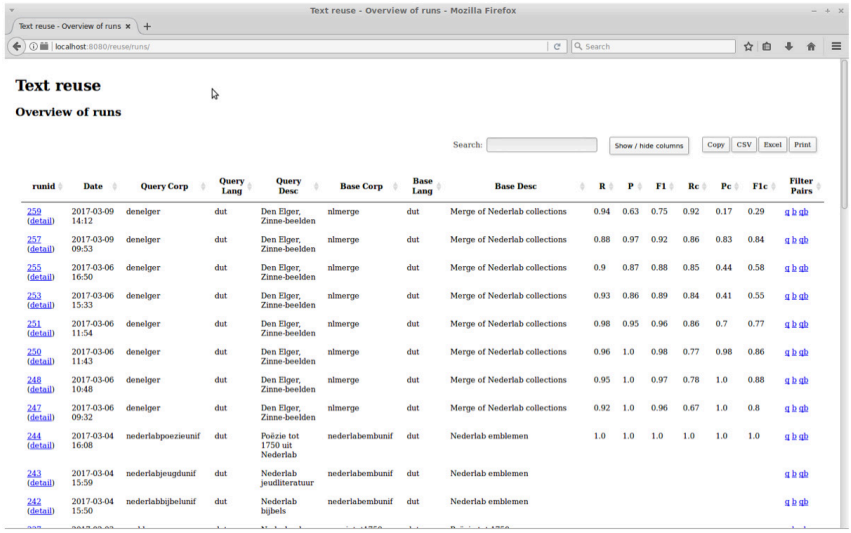

Figure 1. Overview of runs with summary statistics.

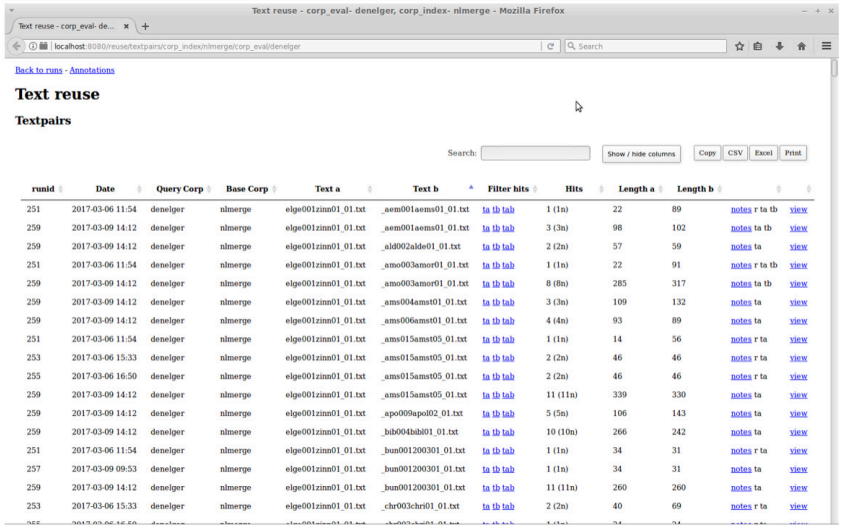

Figure 2. Results for different text pairs in multiple runs.

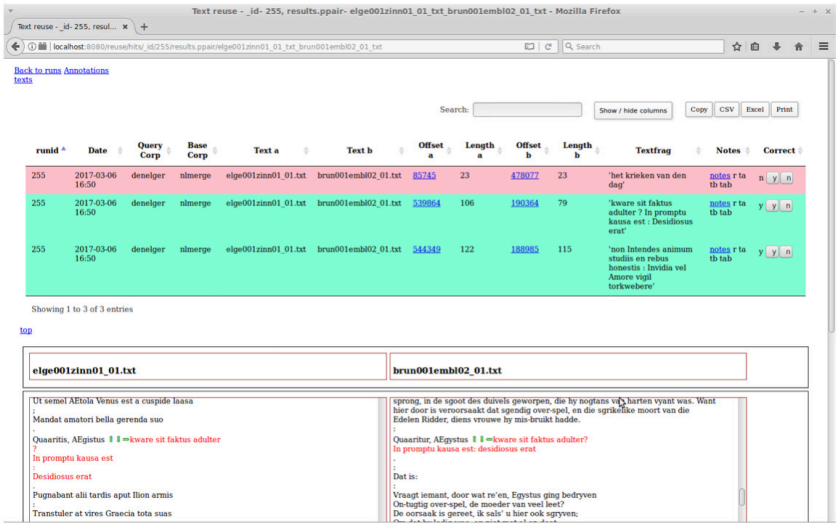

Figure 3. Hits, rejected and approved, for one text pair in one run.

Figure 4. Hits for one text pair in multiple runs.

Armed with this tool, I did an experiment in text reuse detection for the Den Elger book (1703). Table 2 shows the settings I used; the corresponding results are reported in Figure 1. Run 247 was the first run for this experiment. The settings were just a first guess. The parameters are those that were mentioned above, except for the minimum pair of shingles parameter which

gives the number of shingles that the texts should share before an overlap is considered a hit (a single shared n-gram is only considered a hit if the minimum pair parameter is set to 1). The other runs should be seen as attempts to find hits in other parts of the parameter space, without retrieving too many false positives. In run 248 I lowered the required minimum pair of shingles to three; in 250 I set the cut-off size to four. Both increase the number of hits. In run 251, I set the minimum shared pairs to two. This located a number of extra parallels, but also, for the first time, introduced a lot of noise. We see (in Figure 1) that the overlap-based precision is still at .95, but the count-based precision here drops to .70. The difference is caused by the true parallels being much longer than the spurious ones. In response, I increased the shingle size and experimented by no longer filtering out stopwords in run 253. This did not have the desired effect. The remaining runs attempted to locate very short parallel passages (minimum pair of shingles set to 1). In run 259, the last one, I set shingle size to three, increasing the minimum word size to four, hoping to find some three-word quotations. This had the desired effect (count-based recall is higher than in any other run), but only at the cost of producing a large number of false hits. The reported count-based precision of .17 is actually much higher than it should be, as the number of false positives was just too large to annotate all of them.

run	shingle size	min # pair	stopwords	min word size	cutoff size
247	3	4	yes	3	5
248	3	3	yes	3	5
250	3	3	yes	3	4
251	3	2	yes	3	4
253	4	2	no	3	4
255	5	1	no	3	4
257	4	1	yes	3	5
259	3	1	yes	4	5

Table 2. Settings that were used in the experiment on Zinne-beelden der liefde.

Did the management tool actually help me to stay on top of the research data? The answer is: to some extent. I used all the shown functionalities in order to understand the effect of the different settings. I often went back

to the parameter display in order to check which were the parameters for a given run. I used the annotation facilities. The summary statistics quickly became an indispensable tool for judging the quality of parameter settings.

However, the tool also has limitations. On a practical level, the facilities for rejection/approval of hits should be much more powerful. It should for instance be possible to approve or reject all found parallels between two texts in a single action. The auto-annotation feature was sorely missed. It would also be helpful if it were possible to simultaneously display the parallels from multiple runs in the context of the texts.

A more serious limitation is that the tool compares parallels from multiple runs based on offset. It cannot, therefore, compare runs where the input files have undergone some transformation that changes the offsets in the files. For instance, I would have liked to compare the performance of the algorithms on texts with and without spelling uniformization. But as the uniformization changes offsets, it is impossible to look at overlap between parallels in runs with and without uniformization. The same problem would arise if we were to introduce lemmatization, or filtering of text by language or similar operations. There are several ways out of this, but none of them would be simple.

Furthermore, for the count-based statistics, however necessary they are to give due weight to shorter parallels, it is a fundamental flaw that runs that retrieve exactly the same parallel text score differently, if one run splits a retrieved fragment in a number of pieces and another one doesn't.[9] To some extent, decisions based on those statistics must also be flawed.

The last limitation that I want to mention is that the tool forces the annotator to make a binary decision: a parallel is either a true parallel or not.[10] That can be a difficult choice to make, for instance because an undoubted parallel may not be a quotation (either because we recognize the text as a quotation from a third writer or because we don't know whether the older text is the source for the younger text). When Otto Vaenius in his *Amoris Divini Emblemata* (1615) writes "Amor qui desinere potest, nunquam verus fuit" (Love that

[9] For those who are interested, suppose the following: Run1 result (length, offset): 10,5 15,5 20,5 40,10, all approved. Run2 result (length, offset): 10,15 50,10, both approved. Now, for run 1: FN = 1, TP = 4, Rc = 0.8; for run 2: FN = 1, TP = 2, Rc = 0.67 (FN: False negative; TP: True positive; Rc: count-based recall = TP / (TP + FN)). The length-based recall would in both cases be 15 / (15 + 10) = .6.

[10] Actually, the researcher can also refrain from annotating the parallel if the choice is too hard.

can end was never true), at one level he is quoting Jerome, at another level he is quoting his own quotation of Jerome in his earlier book *Amorum Emblemata* (1607–8). Another reason may be that a younger text quotes an older one but inserts a new text fragment within the quotation. Depending on parameter settings, text::pair may ignore the new words and consider the whole text a quotation. But the researcher evaluating the retrieved parallels might want to tag it as "partly correct." The researcher will have to lay down very clear guidelines on when to approve or reject a parallel, but even so, the resulting statistics tell only part of the story.

Summarizing, there is no doubt that a management tool such as this can be a useful extension to a text reuse detection tool. But it is also clear that the tool can suggest a measure of certainty that is not always correct. Especially if quotations are sometimes retrieved in multiple fragments, or the status of a quotation is not quite clear, uncertainties remain. This is unavoidable because of the nature of the research question (maybe most research questions in the humanities are in that respect similar). In the next section, I will look into the question of how this approach to tool usage is applicable to other technologies used in digital humanities research.

Complex tools and complex data in humanities research

In fact, the problems that we encounter in doing text reuse detection are not unique to this problem area or technology. There are many technologies that share similar properties. Verhaar, writing about algorithmic criticism, notes:

> Scholars who aim to compare texts can commonly choose from a broad range of statistical techniques, and it can often be taxing to select an appropriate method. The differences between two distinct classes of texts may be examined using supervised learning techniques, of which Student's t-test, logistic regression and Naive Bayes all form concrete examples. When scholars aim to subdivide a corpus into smaller clusters, they can make use of k-means clustering, calculations of Euclidean distances, PCA or nearest neighbor analyses. These different methods are all based on different algorithms, and they consequently produce different results. Such differences can be subtle in some cases, but also quite dramatic in other cases. Even when scholars have decided to make use of one particular technique, they frequently have the possibility to influence the results by varying some of their initial parameters. In the case of classification, the results of the analyses can often be

> manipulated directly by varying the sizes of the training sets and the test sets. In this study, it was found, for example, that the nature of the network diagram displaying formal similarities between poems can change dramatically along with the variables which are considered. (Verhaar 2016, 207)

To give a number of examples: the popular stylometry tools developed by Eder, Kestemont, and Rybicki (2013) have multiple choices for the features to use in computation, choices for the number of most frequent features to include, multiple distance measures, multiple clustering algorithms, and multiple ways of visualization. In topic modeling, gensim (Řehůřek and Sojka 2010) offers more than fourteen parameters to influence the computation of LDA models. For text categorization algorithms, the same issue is discussed by Koster and Beney (2006).

Parameter settings are hardly ever discussed in the literature. However, bad parameter settings can destroy the efficacy of an algorithm, as shown for instance by Hoste, Hendrickx, Daelemans, and Van den Bosch (2002). It implies that in many cases there is a need for systematic exploration of the behavior of the algorithm in the parameter space. Riedl and Biemann (2012) systematically evaluate the performance of text segmentation algorithms given certain parameter settings, Kievit-Kylar and Allen (2013) do the same for a semantic model of philosophical texts.

The situation for topic modeling is different from the situation for text reuse detection, in that the computation is not deterministic: a second run with the same parameters can result in a different outcome. This makes it even harder to evaluate a single outcome. On the other hand, there exist several measures to automatically evaluate and compare topic models, such as semantic coherence and exclusivity (Roberts et al. 2014).

The situation of a large number of parameter settings resulting in different outcomes creates a number of challenges. The first challenge is the one that we have seen for the text reuse detection case: it is the cognitive burden for the researcher who has to weigh the advantages and disadvantages of many different parameter settings and combinations of settings all producing slightly different results. A related challenge is keeping the administration of all these runs in order: how to make sure that we remember which combination of settings was used to create which outcome or combination of outcomes. This is especially problematic in cases where no ground truth is available. Why should we trust a clustering of authors by their hundred

most frequently used words rather than by two hundred or three hundred? To avoid the human tendency to look for confirmation of our expectations, Eder, Kestemont and Rybicki include in their script the facility to generate bootstrap consensus trees, based on multiple parameter settings. This is an example of an ensemble method (Zhou 2012), methods that combine in some way the output of multiple algorithms or algorithm executions in order to create a better result. Ensemble methods are popular solutions in machine learning applications and other fields. When we try to locate examples of text reuse using multiple runs with different settings, we are applying an informal ensemble technique.

Another situation where the multiplicity of parameter settings and the impossibility to select a single best setting could create problems is when we try to employ tools for computational analysis in the service of exploring a digital collection. Even though practical applications are still relatively rare, much research has been done into interfaces where a digital library can be browsed on the basis of computed characteristics. Chuang, Ramage, Manning, and Heer (2012) describe a topic model-driven tool for exploring a library of PhD theses. Kolak and Schilit (2008) describe the "Popular Passages" tool used to facilitate navigating Google Books on the basis of shared quotations. The Commonplace Cultures project (Morrissey 2016) built a large database of cases of test reuse in eighteenth-century English works. But if there is no single best setting for the parameters of the tool underlying this navigation, the user should be aware that the completeness and correctness (let alone the relevance) of the created links cannot be guaranteed.

How can tools prepare for this?

Given the fact that tools consume and produce data, and that, as we have seen, most problems seem to require multiple tool runs, many research problems easily end up in a small data deluge. I described above how I tried to handle this deluge for the case of text reuse detection. But the question arises: how can we deal with the issue in a more general way? It doesn't seem feasible to require every new tool to come with its own virtual research environment for organizing and evaluating its output. Would it be possible to develop wrappers around tools that store tool input and output? And perhaps a generic reporting and annotation facility around the input and output data, whose application in a certain tool domain should be largely parameter-driven?

It is interesting to note that the requirement of storing information about input, parameters, and output of tool runs is being widely discussed in other scholarly domains and mostly for other concerns than the ones discussed in this essay. Storing provenance information about scientific data is important for a number of reasons, including the need to assess data quality, to provide an audit trail, to provide replication recipes, and to facilitate recognition of individual contributions (Simmhan, Plale, and Gannon 2005). It is also essential for effectively sharing research data (Kowalczyk and Shankar 2011).

A number of tools are being developed to support storing provenance information for scholarly data. I mention two of them. YesWorkflow is a tool for storing and displaying what is called *prospective* provenance (McPhillips et al. 2015). Prospective provenance provides insight in the structure of a script and the steps that are taken to create a certain output file. YesWorkflow requires that the scripts that implement a scholarly workflow are provided with annotations that indicate program blocks, data flows, and connections between scripts. Software can display the information in graphical form, either in a process-oriented or a data-oriented view, and can answer queries about the provenance of an output dataset. In contrast to prospective provenance, *retrospective* provenance is based on capturing runtime information. An example of that approach is noWorkflow (Murta, Braganholo, Chirigati, Koop, and Freire 2014). noWorkflow captures information during the execution of a Python script. What it captures is information about the environment (operating system, environment variables, libraries that the script depends on) as well as information about the execution steps, including function calls (with input and output) and the content of all files accessed by the script.

In the humanities, application of tools like these up to now has been experimental. Senseney (2016) describes an experiment to annotate with YesWorkflow some of the scripts for Ted Underwood's research (Underwood and Sellers 2015). Clark (2012) produces provenance information for XSLT transformations. Clark and Holloway (2012) describe a possible formalization of provenance information in two digital humanities virtual research environments. They stress that storing provenance data is really a traditional humanist virtue: "The humanities as a discipline has traditionally exhibited great care in documenting sources and establishing authentic chains of object transmission," however, "to date, little published research in e-humanities explicitly focuses on data provenance."

In situations where noWorkflow has been used for storing retrospective provenance data, it becomes possible to analyze multiple runs, to look at

their differences, and to re-execute runs with or without changes (Pimentel, Freire, Braganholo, and Murta 2016). This opens the door to using tools for provenance detection not just to understand how the output of a single run was created, but also to understand why output from another run might be different. Bennett et al. (2016) speculate about how provenance data from multiple script runs could be input for machine learning and other data science approaches and thus point to improvements in the scripts.

However, a common complaint about noWorkflow and other systems for capturing retrospective provenance data is that the systems produce very large amounts of data that can easily overwhelm the researcher. In that respect, they may not be the best answer to a problem that is basically one of having to handle too much information. Also, capturing provenance data by itself does not help us in presenting the output of different runs in a way that makes it easy to compare their effectiveness, or in annotation of the results, or in the computation of summary statistics.

The management tool that I described above relied on additional coding added to the scripts that saved the input parameters and the script output into a database. A middle course between this semi-manual approach and capturing complete provenance information would be a management system that allows the researcher to set up runs in a database, by defining parameter settings, defining the files or collections of files to be processed, including metadata describing these files, attaching a script, and selecting the output from the script that should be saved into the database. It should then be possible to execute such a run from the database. The database would then no longer have the status of an afterthought but would be the controlling instance in script execution. A disadvantage might be that the script will have to be written in accordance with the expectations of the management system. This approach begins to resemble scientific workflow systems, such as Taverna or Kepler (Talia 2013); however, in the context of this essay we do not necessarily need their facilities for managing the details of workflow execution, calling web services, or sending jobs to high performance computing clusters.

The desired functionality for displaying and annotating the results and for computing derivative statistics would probably need to be specific to the relevant technology. In our case, lines in the CSV output files correspond to potential parallels. When doing topic modeling, for example, the outcome is a distribution of words over topics and of topics over documents. When doing stylometry, the output of a run might be a dendrogram or a principal

components analysis. The different sorts of output are completely different, and it is hard to imagine a generic tool that knows how to display all of them satisfactorily. But it should be possible to work out an architecture where the general functionality of browsing runs with their input and their output is supplemented by plugins handling specific output and input types. The computation of quality measures such as precision and recall could also be handled in plugins.

It is worth noting that an environment like the one sketched here is very different from the popular Jupyter (IPython) notebook computing environment. When using notebooks, we work interactively. If a step in our computations fails, we change a few things, go back a few steps, run again, and this interactivity has many advantages. Most notably, unlike in the case of a failed script execution, all variables keep their values, and we can continue the computations more or less where the problem occurred. Similarly, after our computation is done, we can immediately continue the analysis based on the outcomes of the preceding steps. In a script execution environment, if we need a further analysis, we add the steps for the extra analysis to the script; we have to run the preceding steps again, and when after twenty minutes our added steps should be executed, they fail because of some small coding error. It is clear that a management system would certainly not be conducive to flexibility. On the other hand, finding out after a few weeks of intensive notebook use which notebook did which computation and which files it produced can be quite challenging. The two approaches probably lend themselves to different situations: the notebook approach to the situation where we are experimenting and still do not know which algorithms to select; the management approach once we have chosen the algorithms and want to explore systematically the best settings. However, for now, given the absence of such a management tool, this remains speculation.

Conclusion

Many questions in the humanities lend themselves to an approach based on or supported by software. Much work in the digital humanities goes into devising algorithms to answer these questions. But when we want to run those algorithms, we have to make choices: choices in the preparation of the input files, in the parameter settings, in the content of a stopword file, and in the visualization of the results. Often, we have no real reasons to prefer one choice over another, and so we try a few settings. But these settings result in different outcomes and often we have no other way of judging those outcomes than our own intuitions. Or, in the text reuse detection case that was discussed in this essay, each of the outcomes may contribute something

to the complete answer, but we don't know in advance what they will contribute and how many tries we will need. Unlike topic modeling, text reuse detection is really simple; understanding the algorithms requires no training in mathematics or statistics. Yet it is only by trial and error that we find the settings that work; most runs find some true parallels and some false ones, and miss other true ones.

In this essay I have argued that in both situations we need software beyond the initial programs devised to tackle the original question. We need that additional software (I called it a management tool) in order to organize, compare, and evaluate the input and output of the initial algorithms. An ingredient in this additional software layer is the facility for unstructured and structured annotation as well as the computation of quality measures for the original results. This additional software may not require the intellectual brilliance of a fundamentally new algorithm; on the other hand, considerations of usability (clarity, flexibility, affordances, response time) become much more important. Maybe the two sorts of program should even be written by different developers.

There is much to be said for trying to develop a generic management system: a large part of the functionality for comparison and evaluation in different technologies should be similar. A generic structure with plugins for specific output types seems most promising. In the absence of such a tool, experimentation with technology-specific tools for managing (storing, viewing, comparing, evaluating, and annotating) the output of scholarly software can teach us what a more general tool should look like. For text reuse detection tools, certainly, it should be possible to define a common output format that would allow the creation of a shared tool for managing the tools' output. For text reuse detection in historic text genres, sensitive as it is to parameter settings, this would certainly help.

This sensitivity to parameter settings is not specific to software applications in early modern studies, but recurrent spelling variation and OCR issues of early modern texts do make the problem particularly urgent in our field. However, the issue also occurs in technologies that are not text-based. A good example is the recent study by Masías, Baldwin, Laengle, Vargas, and Crespo (2017) which uses social network analysis to assess the prominence of characters in Shakespeare's *Romeo and Juliet*. The network construction can be based on different aspects of the play, the centrality of characters can be based on different aspects of the network, and resulting computations are also dependent on a parameter setting. The outcomes are widely different, for instance in how they rate Juliet's importance. Giacometti et al.

(2017) note how results in multispectral imaging of an eighteenth-century manuscript depend on camera, light source, and algorithm. Here, like before, choices that may seem technicalities are in fact methodological choices that should be discussed and evaluated systematically. In order to come to grips with its growing amounts of research data, as well as to clarify provenance and facilitate replication, scholarly research should save and make accessible the complete tool runs on which its arguments are based.

WORKS CITED

Anonymous. 1589. *Een Aemstelredams amoureus lietboeck*. Amsterdam. Accessed 17 February 2020. https://www.dbnl.org/tekst/_aem001aems01_01/.

Anonymous. [1617]. *Nieuwen ieucht spieghel*. No place. Accessed 17 February 2020. http://emblems.let.uu.nl/nj1617.html.

Anonymous. 1637. *Biblia, dat is: De gantsche H. Schrifture* [...] [Dutch States' Bible]. Leyden. Accessed 17 February 2020. https://www.dbnl.org/tekst/_sta001stat01_01/.

Anonymous. 1853. *Een lees- en zangboekjen voor de jeugd*. Amsterdam. Accessed 17 February 2020. https://www.dbnl.org/tekst/_lee015lees01_01/

Bath, Michael. 1994. *Speaking Pictures.* London: Longman.

Bennett, Kristin P., Johm S. Erickson, Hannah de Los Santos, Spencer Norris, Evan Patton, and John Sheehan, et al. 2016. "Data Analytics as Data: A Semantic Workflow Approach." 30th Conference on Neural Information Processing Systems. Accessed 23 August 2019. http://tw.rpi.edu/media/2017/01/10/5104/semanlaytics.pdf.

Bloemendal, Jan. 2007. "Love Emblems and a Web of Intertexuality." In *Learned Love*, edited by Els Stronks and Peter Boot, 1:111–18. The Hague: DANS.

Boot, Peter. 2012. "Similar or Dissimilar Loves? *Amoris Divini Emblemata* and Its Relation to *Amorum Emblemata*." In *Otto Vaenius and His Emblem Books*, edited by Simon McKeown, 157–73. Glasgow Emblem Studies 15. Glasgow: University of Glasgow.

Cats, Jacob. 1618. *Silenus Alcibiadis, sive Proteus.* Middelburg. Accessed 17 February 2020. https://www.dbnl.org/tekst/cats001sile01_01/.

Chuang, Jason, Daniel Ramage, Christopher Manning, and Jeffrey Heer. 2012. "Interpretation and Trust: Designing Model-Driven Visualizations for Text Analysis." SIGCHI Conference on Human Factors in Computing Systems, 443–52. Accessed 17 March 2017. https://www.researchgate.net/publication/254005264_Interpretation_and_trust_Designing_model-driven_visualizations_for_text_analysis.

Cieslak, Katarzyna. 1995. "Emblematic Programs in Seventeenth-Century Gdansk Churches in the Light of Contemporary Protestantism: An Essay and Documentation." *Emblematica* 9.1: 21–44.

Clark, Ashley M. 2012. "Meta-stylesheets: Exploring the Provenance of XSL Transformations." Presented at Balisage: The Markup Conference 2012, Montréal, Canada, 7 – 10 August 2012. In *Proceedings of Balisage: The Markup Conference 2012*. Balisage Series on Markup Technologies 8. Accessed 23 August 2019. https://doi.org/10.4242/BalisageVol8.Clark01.

Clark, Ashley M., and Steven W. Holloway. 2012. "'The Past Is Never Dead. It's Not Even Past': The Challenge of Data Provenance in the e-Humanities." Presented at Digital Humanities 2012. Accessed 23 August 2019. http://www.dh2012.uni-hamburg.de/conference/programme/abstracts/the-past-is-never-dead-its-not-even-past-the-challenge-of-data-provenance-in-the-e-humanities.1.html.

Coffee, Neil, Jean-Pierre Koenig, Shakthi Poornima, Christopher W. Forstall, Roelant Ossewaarde, and Sarah L. Jacobson. 2012. "The Tesserae Project: Intertextual Analysis of Latin Poetry." *Literary and Linguistic Computing* 28.2: 221–28.

Daly, Peter. M. 1998. *Literature in the Light of the Emblem: Structural Parallels between the Emblem and Literature in the Sixteenth and Seventeenth Centuries.* 2nd ed. Toronto: University of Toronto Press.

De Jongh, Eddy. 2000. *Questions of Meaning: Theme and Motif in Dutch Seventeenth-Century Painting*. Leiden: Primavera Press.

De Brune, Willem. 1636. *Emblemata of Zinne-werck*. Amsterdam. Accessed 17 February 2020. https://www.dbnl.org/tekst/brun001embl02_01/.

Den Elger, Willem. 1603. *Zinne-beelden der Liefde*. Leiden. Accessed 17 February 2020. http://emblems.let.uu.nl/el1703.html.

Eder, Maciej, Mike Kestemont, and Jan Rybicki. 2013. "Stylometry with R: A Suite of Tools." Presented at Digital Humanities 2013. Accessed 15 February 2017. http://dh2013.unl.edu/abstracts/ab-136.html.

Forstall, Christopher, Neil Coffee, Thomas Buck, Katherine Roache, and Sarah Jacobson. 2015. "Modeling the Scholars: Detecting Intertextuality through Enhanced Word-Level N-gram Matching." *Digital Scholarship in the Humanities* 30.4: 503–15.

Giacometti, Alejandro, Alberto Campagnolo, Lindsay MacDonald, Simon Mahony, Stuart Robson, and Tim Weyrich, et al. 2017. "The Value of Critical Destruction: Evaluating Multispectral Image Processing Methods for the Analysis of Primary Historical Texts." *Digital Scholarship in the Humanities* 32.1: 101–22.

Heckscher, William S., and Karl-August Wirth. 1959. "Emblem, Emblembuch." *Reallexikon zur Deutschen Kunstgeschichte* 5: 85–228.

Henkel, Arthur, and Albrecht Schöne. 1976. *Emblemata. Handbuch zur Sinnbildkunst des XVI. und XVII. Jahrhunderts.* Stuttgart: Metzler.

Hoste, Véronique, Isis Hendrickx, Walter Daelemans, and Antal van den Bosch. 2002. "Parameter Optimization for Machine-Learning of Word Sense Disambiguation." *Natural Language Engineering* 8.4: 311–25.

Kane, Andrew, and Frank W. Tompa. 2011. "Janus: The Intertextuality Search Engine for the Electronic *Manipulus florum* Project." *Literary and Linguistic Computing* 26.4: 407–15.

Kievit-Kylar, Brent, and Colin Allen. 2013. "Kant Be Understood? Probing the Parameters of Semantic Models of Philosophy." International Association for Computing and Philosophy Conference. Accessed 1 March 2017. http://www.iacap.org/proceedings_IACAP13/paper_36.pdf.

Kolak, Okan, and Bill N. Schilit. 2008. "Generating Links by Mining Quotations." *Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia*, 117–26. Accessed 23 August 2019. https://sites.google.com/site/schilit/kolak_ht08.pdf.

Koster, Cornelis H., and Jean G. Beney. 2006. *On the Importance of Parameter Tuning in Text Categorization*. International Andrei Ershov Memorial Conference on Perspectives of System Informatics. Accessed 23 August 2019. https://link.springer.com/chapter/10.1007/978-3-540-70881-0_24.

Kowalczyk, Stacey, and Kalpana Shankar. 2011. "Data Sharing in the Sciences." *Annual Review of Information Science and Technology* 45.1: 247–94.

Luyken, Jan. 1671. *Duytse Lier.* Amsterdam. Accessed 17 February 2020. http://emblems.let.uu.nl/lu1671.html.

Luyken, Jan. 1685. *Jesus en de ziel.* Amsterdam. Accessed 17 February 2020. http://emblems.let.uu.nl/lu1685.html.

Luyken, Jan. 1711. *Het leerzaam huisraad.* Amsterdam. Accessed 17 February 2020. https://www.dbnl.org/tekst/luyk001leer01_01/.

Luyken, Jan. 1712. *Schriftuurlyke geschiedenissen en gelykenissen.* Amsterdam. Accessed 17 February 2020. https://www.dbnl.org/tekst/luyk001schr02_01.

Masías, Víctor H., Paula Baldwin, Sigifredo Laengle, Augusto Vargas, and Fernando A. Crespo. 2017. "Exploring the Prominence of Romeo and Juliet's Characters Using Weighted Centrality Measures." *Digital Scholarship in the Humanities* 32.4: 837–58.

McPhillips, Timothy, Tianhong Song, Tyler Kolisnik, Steve Aulenbach, Khalid Belhajjame, and Kyle Bocinsky, et al. 2015. "YesWorkflow: A User-Oriented, Language-Independent Tool for Recovering Workflow Information from Scripts." *International Journal of Digital Curation* 10.1: 298–313. Accessed 19 August 2019. http://www.ijdc.net/article/view/10.1.298.

Morrissey, Robert. 2016. "Commonplace Cultures: Mining Shared Passages in the 18th Century using Sequence Alignment and Visual Analytics." *Humanities Commons.* Accessed 19 August 2019. http://dx.doi.org/10.17613/M66369

Moseley, Charles. 1989. *A Century of Emblems: An Introductory Anthology.* Aldershot: Scolar Press.

Murta, Leonardo, Vanessa Braganholo, Fernando Chirigati, David Koop, and Juliana Freire. 2014. "noWorkflow: Capturing and Analyzing Provenance of Scripts." In *Provenance and Annotation of Data and Processes*, edited by Bertram Ludäscher and Beth Plale, 71–88. Berlin: Springer. Accessed 23 August 2019. https://link.springer.com/chapter/10.1007/978-3-319-16462-5_6.

Olsen, Mark, and Russell Horton. 2009. "PAIR: Pairwise Alignment for Intertextual Relations." Chicago. Computer software. Accessed 23 August 2019. https://code.google.com/archive/p/text-pair/.

Olsen, Mark, Russell Horton, and Glenn Roe. 2011. "Something Borrowed: Sequence Alignment and the Identification of Similar Passages in Large Text Collections." *Digital Studies/Le champ numérique* 2.1. Accessed 23 August 2019. http://doi.org/10.16995/dscn.258.

Pimentel, João F., Juliana Freire, Vanessa Braganholo, and Leonardo Murta. 2016. "Tracking and Analyzing the Evolution of Provenance from Scripts." In *Provenance and Annotation of Data and Processes*, edited by Marta Mattoso and Boris Glavic, 16–28. Berlin: Springer.

Porteman, Karel. 1977. *Inleiding tot de Nederlandse emblemataliteratuur.* Groningen: Wolters-Noordhoff.

Potthast, Martin, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. 2010. "An Evaluation Framework for Plagiarism Detection." *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* 997–1005. Accessed 23 August 2019. https://www.aclweb.org/anthology/C10-2115.

Řehůřek, Radim, and Petr Sojka. 2010. "Software Framework for Topic Modelling with Large Corpora." *Proceedings of LREC 2010 Workshop: New Challenges for NLP Frameworks, Valletta, Malta.* Accessed 23 August 2019. http://www.lrec-conf.org/proceedings/lrec2010/workshops/W10.pdf.

Riedl, Martin, and Chris Biemann. 2012. "Sweeping through the Topic Space: Bad Luck? Roll Again!" *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, 19–27. Accessed 23 August 2019. https://www.aclweb.org/anthology/W/W12/W12-0703.pdf.

Roberts, M. E., B. M. Stewart, and D Tingley. 2016. "Navigating the local modes of big data". *Computational Social Science*, edited by R. Michael Alvarez, 51–97. New York: Cambridge University Press.

Saunders, Alison. 2007. "Creator of the Earliest Collection of French Emblems, But Now Also Creator of the Earliest Collection of Love Emblems? Evidence from a Newly Discovered Emblem Book by Guillaume de la Perriere." In *Learned Love: Proceedings of the Emblem Project Utrecht Conference on Dutch Love Emblems and the Internet (November 2006)*, edited by Peter Boot and Els Stronk, 1: 13–32. The Hague: Edita.

Sculley, D., and Bradley M. Pasanek. 2008. "Meaning and Mining: The Impact of Implicit Assumptions in Data Mining for the Humanities." *Literary and Linguistic Computing* 23.4: 409–24.

Senseney, Megan. 2016. "Pace of Change: A Preliminary YesWorkflow Case Study." Center for Informatics Research in Science and Scholarship (CIRSS) Technical Report 201601-1. Illinois: University of Illinois at Urbana-Champaign.

Simmhan, Yogesh L., Beth Plale, and Dennis Gannon. 2005. "A Survey of Data Provenance in E-science." *ACM Sigmod Record* 34.3: 31–36.

Stronks, Els. 2008. "The Emblem in the Low Countries." In *Companion to Emblem Studies,* edited by P. M. Daly, 267–289. New York: AMS Press.

Talia, Domenico. 2013. "Workflow Systems for Science: Concepts and Tools." *ISRN Software Engineering* 2013, article ID 404525. Accessed 23 August 2019 https://www.hindawi.com/journals/isrn/2013/404525/.

Underwood, Ted, and Jordan. Sellers. 2015. "How Quickly Do Literary Standards Change?" *Figshare*. Journal Contribution. Accessed 10 March 2017. https://doi.org/10.6084/m9.figshare.1418394.v1.

Van Leuven, Ludovicus. 1619. *Amoris Divini et Humana Antipathia.* Antwerp. Accessed 17 February 2020. http://emblems.let.uu.nl/ad1629.html.

Van Veen, Otto. 1607–8. *Amorum Emblemata.* Antwerp. Accessed 17 February 2020. http://emblems.let.uu.nl/v1608.html.

Van Veen, Otto. 1615. *Amoris Divini Emblemata*. Antwerp. Accessed 17 February 2020. http://emblems.let.uu.nl/v1615.html.

Valck, Michiel. 1607. Den nieuwen verbeterden lust-hof [...]. Amsterdam. Accessed 17 February 2020. https://www.dbnl.org/tekst/vlac002nieu03_01/.

Verhaar, Peter. 2016. "Affordances and Limitations of Algorithmic Criticism." Unpublished PhD. Leiden: Leiden University. Accessed 19 August 2019. https://openaccess.leidenuniv.nl/bitstream/handle/1887/43241/PhD_PeterVerhaar.pdf.

Zahora, Tomas, Dmitr Nikulin, Constant J. Mews, and David. M. Squire. 2015. "Deconstructing Bricolage: Interactive Online Analysis of Compiled Texts with Factotum." *Digital Humanities Quarterly* 9.1. Accessed 23 August 2019. http://www.digitalhumanities.org/dhq/vol/9/1/000203/000203.html.

Zhou, Zhi-Hua. 2012. *Ensemble methods: foundations and algorithms*. New York: Chapman and Hall/CRC.