Special section on 'Computational Methods for Literary–Historical Textual Scholarship'

Editorial
Introduction to a special section on 'Computational Methods for Literary–Historical Textual Scholarship'

All sorts of surprising discoveries about literary and historical texts have been made in the past 30 years or so by investigators employing new computational methods unavailable to previous generations. One landmark publication was John Burrows's book *Computation into Criticism* (1987), which showed that literary scholars had been simply ignoring most of the available evidence, as expressed in the celebrated opening sentence 'It is a truth not generally acknowledged that, in most discussions of works of English fiction, we proceed as if a third, two-fifths, a half of our material were not really *there*'. Burrows showed that the function words—the 100 or so words that comprise articles, conjunctions, prepositions, and other linguistic 'glue' holding our sentences together—are just as amenable to literary criticism as the more visible, rarer lexical words.

Burrows could undertake his innovative research because digital transcriptions of literary works made it possible to count the function words, and he developed a series of algorithms for processing the resulting counts that are now widely used in the field. Since 1987, many more texts have been digitized and many more algorithms have been invented to process them in various ways. A conference at De Montfort University, Leicester, on July 2018, generously funded by the UK's Arts and Humanities Research Council and by the host university, was an opportunity to take stock of where these three decades of work had brought those interested in analysing texts using computers. This special section of *Digital Scholarship in the Humanities* presents a selection of the best articles from the conference; other fine articles had already been committed to other outlets.

The expansion in digital texts available to investigators, which has occurred in the past 30 years, has come from two means: the keyboarding of existing nondigital texts and the transformation of images of printed pages into digital texts by optical character recognition (OCR) of the letter shapes in those images. The former approach, involving human labour, is several orders of magnitude more expensive than the latter but produces more accurate representations of the original writing. Because it is relatively inexpensive, OCR has been the means by which most of the expansion of our digital text collections has taken place. How much does its inaccuracy matter?

In 'Quantifying the impact of dirty OCR on historical text analysis: Eighteenth Century Collections Online as a case study', Mark J. Hill and Simon Hengchen get a handle on just how good or bad OCR is and how much the badness affects certain applications we put the texts to. They compared the part of the dataset Eighteenth Century Collections Online (ECCO), sold by Gale Cengage, that was manually keyboarded for the Text Creation Partnership (TCP) project with the part that was merely OCR'd, to judge how bad the OCR really is. To determine what difference OCR makes, they ran standard tests in topic modelling, collocation analysis, vector space modelling, and authorial attribution, using the keyboarded and OCR'd versions of the same books.

Digital Scholarship in the Humanities, Vol. 34, No. 4, 2019.
doi:10.1093/llc/fqz071

At the level of individual characters, Hill and Hengchen found that OCR is not especially bad in that it does not register a greatly different number of characters from the keyboarded text, but at the word-type level, it is much worse. Because of shape-recognition errors, there appear to be 2.7 million word types in the OCR corpus compared to about 765,000 word types in the keyboarded corpus of the same books. After taking out certain stop words, the top 500 words in the OCR text and the top 500 words in the keyboarded texts differed by 100 words, when of course they should be the same 500 words. Wholly 6.8%, over 5 million tokens, of the OCR corpus is made of types that do not exist in the keyboarded corpus (which is a kind of false positive) and 0.6% (around 451,000 tokens) of the keyboarded corpus are missing from the OCR corpus (false negatives).

Of course, these type-wise counts tell us when spurious non-words are created by OCR error but not when OCR error creates a wrong word that nonetheless is a real word. Looking at the problem page-wise, Hill and Hengchen were able to estimate the metrics called 'Accuracy' (the sum of True Positives + True Negatives divided by the sum of True Positives + False Positives + False Negatives + True Negatives), 'Precision' (True Positives divided by the sum of True Positives + False Positives), 'Recall' (True Positives divided by the sum of True Positives + False Negatives), and what is called the 'F1 Score' (double the product of Recall × Precision divided by the sum of Recall + Precision), for which they provide the figures. According to Hill and Hengchen, the F1 score is the most useful general measure.

As textual scholars might already expect, looking at the letters that give OCR the most trouble, the letter 's' stands out: it appears in many more of the incorrectly OCR'd words than other letters do. Ligatures, especially those involving 's', also cause OCR error. Long words, of course, give more opportunities for errors to be introduced. Unfortunately, more than half of all word types contain 's' or the ligatures 'ct', 'ff', or 'ffl', so about a quarter of all word-tokens in the corpus are likely to be wrongly OCR'd. Knowing this sort of thing can help those who need to perform automated preprocessing on the texts they want to use from an OCR source, since it directs their attention towards the words most likely to be wrong in that source.

Put to work in different kinds of analyses, Hill and Hengchen found that using the method called Structured Topic Modelling, in which the number of topics is itself inferred from the writing rather than being stated in advance, the OCR corpus produced more topics than the keyboarded corpus (seventy-seven to sixty-five topics) but reassuringly all the keyboarded corpus's topics were present in the OCR corpus's topics list. Moreover, intertopic distance maps show that the topics derived from the OCR and keyboarded corpora are about equally shaped in relation to the words they depend upon. As expected, the topics dependent on words with 's' and ligatures were the ones showing the most difference between OCR and keyboarded texts in the probability distributions that define the topics.

The effect of OCR on word-pair collocation analysis turned out to be more severe: about 490,000 collocations were found in the keyboarded corpus but about 605,000 were found in the OCR corpus and of the 319,000 collocations that did not appear in both lists, 70% were in the OCR corpus. Taking as an example the collocations involving the word 'public' (also spelt 'publick'), Hill and Hengchen found 765 in the keyboarded corpus and 750 in OCR corpus, which sounds good until we learn that fully 305 (nearly half) of these were not in both lists. That is a lot of spurious and missed collocations.

To see the effect of all this on authorial attribution analysis, Hill and Hengchen experimented with three methods—Burrows's Delta method, k-nearest neighbour, and nearest centroid—on single tokens, on token 2-grams, and character 3-grams. The keyboarded corpus was used to make authorial attributions for particular works in the corpus and where the attributions were correct the OCR corpus of the same works was then used to make the attribution to see if it came to the same conclusion. It turned out that corpus size was more important to accurate authorial attribution than accurate OCR was, and indeed sometimes getting a larger corpus made a bigger improvement in attribution accuracy than

getting cleaner OCR did. Hill and Hengchen's conclusion is that for some applications such as topic modelling, even really dirty OCR (below 70% accuracy of words) is still useful. But how do you know how dirty your OCR is? It turns out that for the ECCO OCR'd corpus, the published accuracy rates are pretty reliable and only slightly overstate how good the OCR is.

Also concerned with the digital texts that form the raw materials for our computational approaches is the article 'Agree to disagree: modelling co-existing scholarly perspectives on literary text' by Elli Bleeker, Bram Buitendijk, and Ronald Haentjens Dekker. Whereas Hill and Hengchen were concerned with texts that represent documents as simple strings of alphabetical and punctuation characters—as when texts are encoded according to the ASCII standard or its refinement is known as UTF-8—Bleeker, Buitendijk, and Dekker consider texts to which various kinds of markup encoding have been applied to represent structural and typographical information in source documents. They describe a new attempt to improve on our current markup practices to better represent multiple competing perspectives on a text.

For textual scholarship, the most commonly used standard for markup is eXtensible Markup Language that conforms to the guidelines of the Text Encoding Initiative (TEI). The names of elements—the units of structure and typography—given in the TEI Guidelines are meant to be intuitive and hence to encourage everyone to use them to mean the same things, but in reality, different investigators differ in their understanding of what each element name means and so they do not all use them in the same ways. What is needed is a way to describe what each project means by the element names it uses and for that description to be machine-readable as well as human-readable. This is known as markup semantics, a challenging issue that has been a topic of study for several decades.

Bleeker, Buitendijk, and Dekker intend to achieve this end by expressing information about text in a new data model they call Text-as-Graph (TAG). Rather than representing document information as a tree structure in which every node except the root is a child of a higher node, or as a graph in which nodes are connected by individual lines called edges, TAG uses what is called a hypergraph in which a single edge can join any number of nodes. The markup language for TAG, called TAGML, lets users add multiple layers of markup to the source text transcription. Each layer can express a potentially different scholarly perspective, such as the textual, the documentary, and the rhetorical. Because of the unique structure of a hypergraph, these different layers of markup can overlap. This approach overcomes the well-known problem that markup languages are unable to easily represent overlapping structures, as when the semantic structure of a series of paragraphs allows the user to create a paragraph that overruns the bounds of an alternative way of looking at the text, which is a series of pages.

To accompany their innovative markup technology, Bleeker, Buitendijk, and Dekker developed a workflow, based upon the version-control system called Git, which allows many people to work on the same text at once without needing a purpose-built multi-editing platform. (There already are quite a few of those and none shows any sign of being the one that everybody will adopt.) This workflow requires a complex comparison of the TAGML files, since these may differ on the level of the text as well as on the level of the markup. Bleeker, Buitendijk, and Dekker's article explains how they expand standard collation principles—developed for collating base transcriptions—to the collation of differing markup of the same base transcription.

Once we have texts to work on, there are ever-expanding ways of applying them to literary–historical questions we want to answer. One of the earliest computational applications, preceding even the widespread availability of digital computational services, is authorship attribution: figuring out who wrote a text by analysing what is in it. In 'Finding ''Anonymous'' in the digital archives: the problem of *Arden of Faversham*', Gary Taylor presents compelling evidence that the early modern play *Arden of Faversham*, first published in 1592, was written at least in part by the relatively obscure author Thomas Watson. (At the opposite end of the spectrum of fame, William Shakespeare has recently been

convincingly claimed as the author of at least the central part of this play, its third act.)

Arden of Faversham is especially hard to attribute because it comes from the period before 1594, for which we have fewer plays than later on and fewer still that are single-authored and securely attributed. Adding to the problems, *Arden of Faversham* probably has at least two authors and if one of them was Shakespeare then it was not the Shakespeare we know from his later, more mature, works but Shakespeare near the beginning of his career. Moreover, some candidate authors for writing *Arden of Faversham* left us no single-authored securely attributed plays, such as Thomas Achelley, Michael Drayton, Richard Hathway, and Thomas Watson.

Taylor constructed a corpus of fifteen dramatists' complete output from Early English Books Online Text Creation Partnership (EEBO-TCP). He took as his sample from *Arden of Faversham* the run of thirty-four lines from 10.1 to 10.34, being 274 words from which he extracted every 2-gram, 3-gram, and 4-gram and searched for them in EEBO-TCP. He also searched for 'every collocation of two or more semantic words (nouns, verbs, adjectives, and adverbs) ten words before or after each other'. He was looking for matches that appear only in *Arden of Faversham* and the canon of just the fifteen candidate authors. Taylor counted maximally, so each subgram within a 3-gram (of which there will be two 2-gram subgrams) or a 4-gram (of which there will be two 3-gram and three 2-gram subgrams) added one more to the count of matches, but he also tallied what would be the result if we counted only the largest *n*-gram as a single match and ignored all the subgrams within it.

Looking in the dramatic canons and adjusting for canon size (of which Shakespeare's is the largest), Thomas Watson came out as having the most matches with *Arden of Faversham* 10.1–34 and by a huge relative margin. Turning to the non-dramatic canons, the same thing happened: Watson had by far the most connections to *Arden of Faversham* 10.1–34. In both approaches, Thomas Kyd (Brian Vicker's preferred candidate) came second to Watson. The same thing happened when Taylor counted only works from 1585 to 1594. Next Taylor tried weeding out the links that are unique to one of his fifteen candidates but no longer unique when we consider all of EEBO-TCP from 1585 to 1594. This did not add to his candidate list, but it did remove phrases that are not, in fact, peculiar to one writer, and the result was still the same: Watson came out on top, whether Taylor considered only dramatic canons or canons in all writing genres. Whatever way you slice it, Watson seems to have written *Arden of Faversham* 10.1–34.

Also concerned with authorship attribution is David Hoover's 'Simulations and difficult problems', and as well as tackling particular unsolved problems he considers how our methods of detection might be improved. One of his examples is the same *Arden of Faversham* that Gary Taylor considers. Hoover addresses the problems arising when the disputed text is itself rather shorter than we would like, when we have too few well-attributed texts to compare with the disputed text and are uncertain about their genres, and when the well-attributed texts are co-authored and we do not know who wrote which part. Hoover illustrates the making of what he calls 'simulations' in which we treat a well-attributed text as if it were of disputed authorship and experiment to see what our tests say about it to reject those tests that fail to attribute this text to the person we know wrote it.

A claim was made in the early 2000s that a group of anonymously and pseudonymously published short stories were the undiscovered early works of Henry James, and Hoover became involved in trying to confirm the attribution. The samples were small and we know that James's style changed over time and we have few early works to go on. Hoover recreated the tests that had been used to make the attribution and found them to be essentially unsound when he ran a simulation. It was demonstrable that the tests were particularly sensitive to sample size and liable to give false attributions to larger texts.

Hoover considers the authorship of the anti-Mormon book *Female Life among the Mormons* (1855) which has been attributed to Cornelia Ferris, wife of Utah governor Benjamin Ferris, and a couple of related cases. His procedure is to model a series of scenarios—such as all the books are by

the same author, each is by a different author, and so on—and for each one, he determines using Student's *t*-test how likely it is that he would have got the results he did if this scenario's premise were true. He then adds some known-to-be true scenarios and known-to-be-untrue scenarios about other works and compares the results he got for those with the results he got for the scenarios in dispute. The result is that Ferris writing *Female Life among the Mormons* is especially unlikely.

Hoover turns to some collaborations of Robert Louis Stevenson and Lloyd Osbourne, for which we have Osbourne's account of who wrote which part and some corroborating accounts from Stevenson. Hoover investigates these accounts of who did what, using some of the tests build into the Stylo package for the R statistical computing environment. Again, he creates what he calls simulations as validation runs using cases of known authorship set up with the same parameters as those applied for the experiment. In some cases, he has to recreate the known circumstances by stitching together parts of Stevenson's and parts of Osbourne's writing to match the hypothesis about collaboration that is being tested.

Hoover returns to a topic he has addressed before: Brian Vickers's claims about early modern drama and the utility of *n*-gram matches between words. Hoover points out that Vickers's insistence on looking only at plays from around the same time as the one he wants to attribute risks overlooking valuable evidence from a later period if the true author was much younger than Vickers's preferred candidate. Moreover, Vickers never calculates just what results—how many shared *n*-grams—we should expect to find if his candidate is not the true author. Hoover here extends his previous work replicating Vickers's methods but using nineteenth-century drama instead of early modern drama. His conclusion is that Vickers's method used to attribute *Arden of Faversham* to Thomas Kyd would misattribute the plays Hoover is testing it with.

The point of Hoover using the Victorian drama is that it provides enough plays to do the kind of simulation that is not possible with the early modern drama because, for example, Kyd's securely attributed canon has at most only three plays. Taking the validation still further, Hoover invents pseudo-authors by splicing together parts of different writers' plays, and these pseudo-texts also get seemingly conclusive attributions by Vickers's method, which of course they should not. Hoover concludes that shared rare *n*-grams alone do not provide adequate evidence of shared authorship.

Authorship is not the only aspect of language that computational approaches can shed light upon. In 'A computational approach to lexical polysemy in Ancient Greek', Barbara McGillivray, Simon Hengchen, Viivi Lähteenoja, Marco Palma, and Alessandro Vatri show that we can get some sense of the changing meanings of certain words across time and by genre using automated methods. Words' shifting meanings are of interest to scholars studying writings from the past who want to be sure that they are assuming the historically correct meanings for the words they study. McGillivray *et al.* take a probabilistic view and use Ancient Greek semantic change as their subject. One approach is to look at the company a word keeps: the context will help you decide which of its connotations is or are currently active. That works for a synchronic analysis but how can this be done diachronically?

The team took an automatically lemmatized corpus of 820 texts of Ancient Greek writing, comprising around 10 million tokens, ranging from the time of Homer to the fifth-century AD. First, they annotated by hand, using their knowledge of the historical meanings, the sentences containing the three polysemous words 'mus' meaning mouse or mussel or muscle or whale, 'harmonia' meaning fastening or agreement (harmony) or stringing (musical scale and melody), and 'kosmos' meaning order or decoration or world. (Actually, for the last one, there were too many occurrences so they looked only at the ones before 142 AD.) The annotation specified the meaning they thought active in each occurrence and whether they had determined this by collocates in the same sentence, or by knowledge of the wider text, or by common knowledge of the world, or by logic, and, or by the text's genre or register. Typically, automated systems only decide by the first of these means, the word's collocates in the same sentence.

This manual annotation gave the investigators the data to explore how the various senses of each word that were active in particular kinds of texts at particular times varied across time and also (since we know the genre for each text) to see how genre figures in this change. That is, if a particular sense of a word seemed to become more prevalent as a particular genre of writing became more prevalent, perhaps the reason is that this is the sense of that word that predominates in that genre. Tracking this possibility, the team found that genre did indeed seem to play a role in the particular sense of each word that is active in each use.

It is also possible to track innovations in sense, since if a word is present in a genre in one sense and later appears in that genre in a different sense, this is not a case of the genre shaping the sense but of time doing so. Of course, the new sense might have already been active in another genre before it appeared in this one, and the investigators tracked that. McGillivray *et al.* compared their handmade analysis with an existing computational model that attempted the same sort of thing, using the above example words and also some words for which independent research by others has sketched the changing polysemy.

Ancient Greek writing is a fairly small field within the whole of textual studies, and in 'Beyond digitization? Digital humanities and the case of Hebrew literature', Itay Marienberg-Milikowsky makes the case that in even smaller fields those employing computational approaches simply cannot afford to work in isolation from their non-computational colleagues. In the small world of modern Hebrew literature, he argues, there simply are too few colleagues to leave anyone out. By modern Hebrew literature, Marienberg-Milikowsky means the works written since the mid-nineteenth century. Unlike in other traditions, there are scholars who have read all the works in this one, so we can compare the results of distant reading to the results that we already have from close reading. Why are there literally no computational analyses of modern Hebrew literature? One reason Marienberg-Milikowsky identifies is that the field is obsessed with preservation and accessibility and gives these practices more credit than it gives analysis. Marienberg-Milikowsky's suggestion for overcoming this is that those using computers should talk within his discipline about what computational methods might be able to do that more traditional scholars actually want.

Giving a concrete example of what computational methods offer that non-computational scholars might actually want is our final article, 'Quantitative measures of lexical complexity in modern prose fiction' by Ewan Jones and Paul Nulty. They show that computers can detect one particular literary feature, commonness of lexical choice, that turns out to be more important to criticism than we might otherwise think. One of literary writing's defining characteristics is lexical complexity, but can we quantify it? Sentence length has long been treated as a marker of complexity, for example, in the widely used Flesch-Kincaid Readability index, which also uses syllable counts. Jones and Nulty survey some other 'difficulty' measures that also use sentence length, before turning to other measures such as the relative proportions of various parts of speech such as adjectives and adverbs.

Another approach to difficulty is to measure 'abstractness' by first rating a collection of words according to where they appear on a scale from 'most concrete' to 'most abstract'. Jones and Nulty plot where various literary works of the nineteenth and twentieth centuries appear by the various measures, and the results do not much correspond to conventional literary wisdom. For example, James Joyce's *Ulysses* rates about as concrete a Robert Louis Stevenson's *Treasure Island*. Moreover, Joyce is about as concrete in the collection *Dubliners* as he is in the experimental *Finnegans Wake*, showing that concreteness is not really a marker of what readers report as the subjective experience of textual complexity. Jones and Nulty go on to show that no previously existing measure of complexity accurately tracks the characteristic features that literary scholars mean by complexity, so they created one that takes in 'the historical–cultural context in which a work is embedded'.

Jones and Nulty consider Henry James's corpus to be particularly useful in this regard for two reasons: his style is, on various objective measures, unlike other writers' styles, and moreover his style

changed markedly over his career. They distinguish between 'difficulty' and 'readability', the former being somewhat historically conditioned. That is, Shakespeare's works are harder for today's readers than their first readers because the language has changed but also in certain ways they are easier because some of his innovative uses of language have become for us cliches. To get a handle on this, Jones and Nulty use the kinds of tests that isolate the words that are distinctive of a particular author (such as John Burrows's Delta and Zeta tests) but with whole extracts from Google's Books Fiction dataset providing the background against which this distinctiveness is measured.

We can count how often a word gets used to get at a sense of how familiar it would have been to readers, but Jones and Nulty point out that raw word counts are misleading since they depend on corpus size. Moreover, relative counts (raw counts divided by corpus size) are also imperfect since they too follow a 'power law scale that does not map directly onto a linear score for human judgment of familiarity'. Jones and Nulty prefer the measure called Standard Index of Frequency using a logarithm transform, which has been shown to more accurately reflect just how familiar words seem to people, as judged, for example, by how long people take to make sense of them. The particular calculation they use is as follows:

$$\log_{10} \text{ of } \frac{\text{word-count} \times 1{,}000{,}000{,}000}{\text{the number of tokens}}.$$

Thus, if the word 'apple' appears 160,000 times in a corpus of 7.1 billion tokens, the calculation is $\log_{10}(160{,}000{,}000{,}000 \div 7{,}100{,}000{,}000)$ or $\log_{10}(22{,}535) = 4.35$. The unit for this calculation has been named the Zipf. To apply this to a text, Jones and Nulty calculate the above for each token in the text, excluding those that do not appear in the reference corpus, and take the mean for all the tokens.

Plotting a series of nineteenth-century books on this new Zipf scale separates them more or less as we would expect—Joyce stands out, for example—but it is important to realize that a book using a lot of words less frequently than is usual in the period will make it stand out here just as much as if it used a lot of words more frequently than is usual. We find that James uses more words that are common in the language than many other writers do, including some writers of children's fiction, and that he does this even more noticeably towards the end of his career when he is traditionally supposed to have got harder. So good is this measure of difficulty that Joyce's *Finnegans Wake* had to be omitted from the investigators' chronological visualization plot as it was such an extreme outlier that it would have altered the scale and rendered other works' finer distinctions invisible.

If we go looking for particular sentences that are most typical of sentences of their time, James's writing is not so commonplace as Samuel Beckett's, but when we move up to the paragraph, James becomes the dominant user of commonplace language. Jones and Nulty cite a passage from James's *The Golden Bowl* that is exceptionally typical in its lexical choices yet they find it remarkable. In its context within the novel, Jones and Nulty read the passage as exceptional in using banal language because the character speaking cannot find words that rise to the level of her emotional intensity. Across a series of novels, Jones and Nulty find this happening: the language becomes most commonplace at moments of great importance in the plot or of high emotion. That, they insist, is a literary-critical insight.

The investigations presented here show the variety of ways in which computational methods are shining light on topics that non-computational textual scholars care about, and they prove that the new techniques and methods can be articulated perfectly well without recourse to highly specialist language. As such, they illustrate the possibilities for dialogue across the textual disciplines that bring in the widest possible range of investigators in shared endeavours that make the best of the oldest and the newest approaches to writing. They are presented here by the guest editor in the hope of furthering interactions across the field of textual studies that bring together disparate methods and those practising them.

Gabriel Egan
School of Humanities, De Montfort University