"Impalpable hits: Indeterminacy in the searching of tagged Shakespearian texts" by Gabriel Egan

Abstract In Shakespeare studies, as in the rest of early modern literary studies, the new information technologies have been neither rapidly nor effectively adopted in research. One reason is a misplaced attention upon the notion of hypertext and the seeking of spurious analogies with the early modern printed codex. This essay is concerned with machine applications of textual searching technologies, which is where we should be focussing our energies, and it argues that important recent products for Shakespearian research are weak, and more importantly non-standard, in their searching mechanisms. The desirability of adopting an existing standard, called 'regular expressions', is argued.

Perhaps participants in this seminar were, like me, excited a few years ago when amongst the first new collections of essays in the field of early modern literary studies to appear after the turn of the millennium was a volume called The Renaissance Computer: Knowledge Technology in the First Age of Print (Rhodes & Sawday 2000b). It is a fine collection of essays in many ways, and contributor after contributor makes the apposite observation that just as 400 years ago manuscript culture was, to use Raymond Williams's useful categorization (Williams 1977, 121-27), 'residual' while print culture was 'emergent', so we are at a point where print culture is becoming 'residual' and electronic culture is 'emergent'. One culture did not (and presumably will not) entirely displace the other, but rather the two overlapped for some time as key activities moved to the newer one. Contributors to the volume made much of the fact that the modern reader's experience of hypertext could be likened to the medieval reader's experience of a sophisticated, multi-layered manuscript (Rhodes & Sawday 2000a, 12), that Ovid's Metamorphoses itself, as well as its inset stories, are hypertextual (Brown 2000), and that Thomas Heywood's Gunaikeion is structured like a hypertext in contradistinction from masculinist textual linearity (Crook & Rhodes 2000).

This focus on hypertext--indeed this conflation of the new etext technologies with hypertext--is typical of our field and draws at least part of its strength from George P. Landow's landmark publication Hypertext: The Convergence of Contemporary Critical Theory and Technology (Landow 1992) that took the lightest aspects of high French literary theory and observed that electronic hypertext seems to make real what Roland Barthes described as the ideal state of writing, and that it seems to provide intensified intellectual experiences that find echoes in the works of Jacques Derrida and Michel Foucault. Landow himself traced hypertext back to the essay "As we may think" by Vannevar Bush, which described an imaginary machine (the memex) for recording one's researches through academic books and which many have claimed marked a new direction in theories of how knowledge is organized. This is Bush:

> Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing. When data of any sort are placed in storage, they are filed alphabetically or numerically, and information is found (when it is) by tracing it down from subclass to subclass. . . . The human mind does not work that way. It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in

> accordance with some intricate web of trails carried by the cells of the brain. (Bush 1945, 106)

Bush's memex was a mechanical means of recording the conceptual trails that one's mind creates between items one had read and Bush has been hailed as the inventor of hypertext, itself supposed to be a new form of textuality.

This, I suggest, is a mistake. Writing in 1992 Landow experienced hypertext primarily in the form of the Apple Macintosh application HyperCard, and HyperCard stacks (the trails of connected items) were for the most part written by intelligent people whose brains were worth crawling through. Since then, the proliferation of hypertextually linked documents on the Worldwide Web has proved beyond any doubt that most brains are not worth crawling through and that a significant minority of them are highly objectionable. The problem started, I suggest, with Bush's apparent rejection of indexing in favour of associative linking of disparate materials. As Jim Whitehead pointed out (Whitehead 2000), Bush's notion of an associative link between documents was vague and its implementation in hypertext systems is usually simplistic: the 'head' of a link appears in one document (and is indicated to the user by a visual feature such as underlining) and it leads to the 'tail' located in another, reached by selecting the 'head'. This impoverished notion of association between documents was simple to implement in HyperText Transfer Protocol (HTTP) and HyperText Markup Language (HTML) and does not do justice to Bush's subtle, but vaguely defined, sense of joining two documents.

Such an impoverished sense of hypertext suffuses the writings in Rhodes and Sawday's collection The Renaissance Computer. Thus Thomas N. Corns's essay "The early modern search engine: Indices, title pages, marginalia and contents" defined hypertext as "the linkage of files, sometimes containing very disparate kinds of material, that facilitates movement between those files in ways that are intended to be illuminating" (Corns 2000, 96). There is a distinct ambiguity in the syntax here, for strictly this is a claim that movement between documents happens in ways that are illuminating, which is untrue: in all the systems he described the movement can happen in just one, dull way. That is, the second text opens up instead of, or alongside, the first. Corns really meant that the conceptual similarities of the two documents are interesting, but this is a feature of the texts themselves, not of the technology that links them, and Corns was strangely insistent that the technology itself makes Renaissance texts like hypertext: "The analogy of pointing and clicking makes the first bridge, conceptually, between searching electronic media and non-serial access to early modern texts, which is the subject of this essay" (Corns 2000, 96). Corns started with the 1611 edition of Coryate's Crudities, which he claimed has a "neatly turned graphic interface" in the form of its title-page in which several vignettes are labelled with letters of the alphabet that link each represented incident with an epigram in the front matter of the book. One of the epigrams refers to another, and, by the movement of following such a link, the title-page picture and the epigrams (and, it is implied, the narrative of the book) "are juxtaposed in effect hypertextually" (Corns 2000, 97). In fact there is only one such hyperlinked epigram, but more importantly the alphabetic labelling that connects picture to words is entirely without significance (the letters are arbitrary symbols) and this is in effect a map legend spread over several pages and not hypertext at all, not even the simplest kind. Coryate's Crudities is doggedly old-fashioned in its non-seriality (the feature that Corns thought makes it ultra-modern), as is clear from the acrostic of Coryate's name (Coryate 1611, b4), which is a truly ancient way of making a single word the random-access (non-serial) key to a collection of lines of verse.

Corns's subsequent examples of what he saw as Renaissance hypertextuality are as false as the first, and lest it be thought this assessment ungenerous I should make clear that Corns's literary scholarship in this essay is beyond reproof. The problem is that he wants the early modern texts to be proto-hypertext--emerging with the new textual technology of print--when the features he examines (glossarial annotation, non-seriality, internal linking) were already long established in manuscript culture. That we are in a transitional phase between two textual technologies, just as writers in England were 400 years ago, is undeniable and to point out the structural analogies is valid. It is a fault, however, to see in the works from that time the beginnings of modern forms of textuality that have been driven by technology. One does not have to be a Marxist (though it helps) to accept that changes in the structure (or, base) of production have indeed driven changes in the superstructure, in cultural production as elsewhere. Where Corns could find nothing resembling hypertext in the sample literary works, he settled for mere non-seriality indicated, for example, by the illustrations and marginalia in George Puttenham's The Arte of English Poesie. With his consideration of Charles 1's Eikon Basilike he gives up on this faint technological analogy to observe that in form the book is "a neat, palm-top octavo at a time when most political tracts are laptop quartos" (Corns 2000, 101). Corns's final example, the Geneva bible, is one of those "desktop folios" that are so heavily annotated that the reader's freedom of interpretation is distrained, a process of textual coercion that Corns mistakenly fretted would be made worse by electronic hypertext (Corns 2000, 102-03).

Corns's essay typifies a trend in themed anthologies of essays: the shoehorning of perfectly good scholarship into an unsuitable container. Another recent example is Jean E. Howard and Scott Cutler Shershow's Marxist Shakespeares (Howard & Shershow 2001) in which respectable and scholarly non-Marxist work was sprinkled with spurious remarks about commodity fetishism and reification to suit the collection's title. Surprisingly, although Corns had virtually nothing to say about indexing, the index to the collection in which his essay appeared has 9 entries for the word 'hypertext' and only one for 'index', and that one points the reader to the full span of Corns's essay. This must be because Corns's title includes the word 'indices', and one of Corns's illustrative texts, Michael Drayton's Poly-Olbion, has a rudimentary alphabetized index. However, Corns's was scathing about Drayton's index and wrote nothing further on the subject (Corns 2000, 99). Contrary to the view promulgated by Corns's essay and by the book as a whole, it is not hypertext--and certainly not hypertext in its attenuated sense--that distinguishes technology's impact on writing. Non-serial reading was, as Jonas Calquist recently showed, a use of their work that medieval manuscript writers anticipated and aided (Calquist 2004). The important development in textuality that characterizes the computer age is the automated generation of indices, alphabetized and otherwise sorted. In medieval manuscript culture one finds the occasional index, and in early printed books too: Thomas Cogan's The Haven of Health (Cogan 1584) has one in order that it can like its modern counterparts be consulted rather than read serially. It is worth noting that the new technology of movable type offered no benefit here: indexing remained a dull, manual task, and it is a great irony that we all have access to computers more powerful than the one that Marvin Spevack used to make a full-text index (that is, a concordance) of the Riverside Shakespeare and yet most of us index our own publications by hand. Rather than asserting spurious parallels with the new technology of the Renaissance, humanities scholars ought to stress those benefits we have that were previously unavailable to scholarship. The full value of the latest electronic tools is obscured, however, by the awkwardness of the front-end software that we are forced to use.

A good example of what was, for practical purposes, previously impossible is the searching of a dictionary by definition rather than by the word defined, as we can now do with the electronic versions of the Oxford English Dictionary. Equally, because the full texts are indexed, the combined resources of the English Short Title Catalogue and Literature Online allow one to say with confidence that not only was the 1609 quarto of Wilkins's and Shakespeare's Pericles the first time (at least since the mid-sixteenth century) that a drama was described by the word 'play' on its title-page, but this was also the first play to use the new word 'title-page'. The underlying data structures of the big databases that Shakespearians consult are generally formed using well-established and essentially open (as opposed to proprietary) standards. Chadwyck-Healey's etexts have always been tagged using Simple Generalized Markup Language (SGML) and newer products often use eXtensible Markup Language (XML), both of which are standards of the International Standards Organization (ISO). Products based on well-established database technologies are generally well-structured enough that direct access can be made using Structured Query Language (SQL), another ISO standard. Unfortunately, this relative uniformity 'under the hood' is not matched by uniformity in the front-end software encountered by a user of these products, and each software supplier has chosen to invent its own conventions for how searches are constructed. This is a shame, because we all want to do essentially the same things with these products, namely:

search for a string of alphanumeric characters that can occur anywhere in a database record;
substitute one or more wildcard characters (signifying any character in a range) for one of the alphanumeric characters in the above;
combine search expressions using Boolean logic (and, or, exclusive-or, and not);
combine searches using proximity operators so that the found terms have to be within a specified textual distance of one another.

I have yet to find two products from different suppliers in which the above activities were accomplished using the same conventions and symbols, which means that scholars using these systems must memorize a set of mutually incompatible habits for accomplishing what is essentially a single task.

To give a sense of how much variety exists in the ways the above tasks are executed on electronic databases that Shakespearians routinely use, I will confine myself to the advanced searching features of the current (as of January 2005) versions of the World Shakespeare Bibliography (WSB), the English Short Title Catalogue (ESTC) web-based and CD-ROM versions, Early English Books Online (EEBO), Editions and Adaptations of Shakespeare (EAS) CD-ROM, Literature Online (LION), and the Oxford English Dictionary (OED) on CD-ROM . I am not mainly concerned with (but will mention in relation to ambiguity in general) the specific ways that fields can be tied together for the purposes of searching--about which every front-end designer also seems to take a different view--because necessarily this can be conditioned by the natures of the fields in the records, and these vary from product to product. I will confine myself to wildcarding, Boolean logic, and proximity connectors. Many database products (for example, the World Shakespeare Bibliography) offer, as an alternative to searching, a feature called browsing in which predeterminedly useful subdivisions of the data and ordering of items within the subdivisions are applied without consulting the user, and the results are presented in list form. This is the closest that electronic resources get to a codex's arrangement, and indeed is merely a hangover from that form. Once data are stored in a properly relational structure there is no such thing as merely browsing the records: every browse is in fact a search with preset criteria and sorting of the results.

The World Shakespeare Bibliography has, in its quick search function, a set of what are called radio-buttons--as in old motor-car radios, selecting any one item deselects the item previously chosen--for the fields author, title, and year; thus only one of these fields can be searched at a time. The box into which a search term is entered is called in WSB's terminology (and in many American products) keywords, which is where a distinct ambiguity creeps in. The OED's sense of a keyword is "any informative word in the title or text of a document, etc., chosen as indicating the main content of the document" (OED key n.1 18) and British writers tend to confine themselves to this sense. Thus Raymond Williams's descriptive index of important cultural and political terms that had changed their meanings in significant ways over the centuries, written for but omitted from his book Culture and Society (Williams 1958), was published separately as his book Keywords (Williams 1976). However, the WSB's quick search and its advanced search features use the word keywords in an entirely different sense: as a pseudo-field comprised of an amalgamation of the fields author, title, publisher, notes, reviews, and people, so that whatever is entered as a search of the keywords pseudo-field will find a match with the corresponding text in any of the real fields it stands for. This second sense of the words keywords could stand as a distinct alternative to the British sense, were it not that the WSB's help system uses keywords in yet another sense, meaning a word appearing anywhere in a field, which is a common American usage and equivalent to free-text searching in some terminologies. It is no wonder that users are often baffled by the help systems that come with products, and while they usually are able to construct a search that they are confident will find all that they are looking for, they are usually unable to define a search that will definitely exclude what they do not want to find. That is to say, these technologies are ambiguous at the cost of sacrificing determinacy in searching.

In WSB there is no obvious hint how to form an author's name for the purpose of searching in the author field--or indeed, the keywords field, which will hit an individual's reviewing outputs as well as her primary authoring--but the software appears to do some clever reordering if two words are entered. Thus searching within keywords for the author gabriel egan hits egan, gabriel as well as gabriel egan and so returns the same 54 records as a search for egan, gabriel. However, this does not work if 3 terms are entered: gabriel i egan misses all but 2 of the records and it is not clear why, since the automatic reversal of first and last name still takes place (one hit is gabriel egan, the other egan, gabriel) but possibly the reason is that the found records have to contain the single character i, which is rare in bibliographical descriptive prose. In both hits a match was found with the i in the HTML tag <i> used to denote italicized text, which tag is normally hidden but which this search makes visible. There is a particular irony in an American product working well with the 2-part names that most British scholars have and failing with the 3-part names (including a middle initial) that most American scholars possess.

In programming logic, and indeed in life generally, there are two kinds of logical or. The inclusive or is the familiar one of being allowed, say, fruit or juice for breakfast (you may have both) while the exclusive or applies in situations where choosing both is not allowed: coffee can be served white or black. None of the software described here acknowledges the existence of the exclusive or in its documentation or its programming implementation, so the WSB is merely typical in this failing. Although it might seem odd to want exclusive or (sometimes written as EOR or XOR) to be included in implementations of Boolean logic, it can be the most succinct way to describe what one seeks: Laurel EOR Hardy would find only their rare solo works and is more succinct than (Laurel NOT Hardy) or (Hardy NOT Laurel). Moreover, it avoids the need to bracket terms, a subject about which the help systems of the products surveyed are especially coy. Another area of potential confusion (in addition to the problem with the multiple meanings of the word keywords) is wildcarding or truncation. The idea is that a special symbol found on the computer keyboard (usually a question mark or an asterisk) represents any character in the alphabet (and perhaps any digit too), and so is wild in the sense that aces and jokers can be wild in the game of poker. Some systems allow the wildcard to stand not only for a single character but for a whole string of characters. The alternative name of truncation for this feature arises because in certain systems (and WSB is one) the wildcard (in WSB the asterisk) is allowed only at the end of a search word, and hence it is as though the search term were a truncated stem permitted to match all words that begin with its set of characters. Thus wom* would hit (amongst many others) womb, woman, and women.

Regarding this feature the products' help systems are occasionally misleading, as when WSB's documentation claims that the asterisk is "used to represent one or more variable characters at the end of a search word". In fact it represents zero or more variable characters at the end. Were it truly "one or more" then a search for lear* would hit learn and lears but would miss lear itself, since the software would be looking for words of at least 5 letters (the 4 of lear plus "one or more"). A WSB search for lear* does in fact hit lear, so we can be sure that the asterisk is allowed stand for zero or more characters, which indeed is its usual sense in the programming languages that underlie the software used. I will return to this point in my conclusion. The WSB does not allow the wildcard to appear within the search word(s), so le*r would be disallowed as a search, and hence it has no way to represent just a single wild character, as one would want in order to search for leir and lear by making just the 3rd character wild. In the current version of WSB, proximity criteria--that term A must occur within a certain number of words of term B--cannot be specified and the help system is silent about this familiar searching feature. (One area in which the documentation produced by technical writers almost invariably surpasses that written by textual scholars is that of the owning up to limitations. To save someone wasting time looking for it, a technical writer would insert into the documentation a section called Proximity Searching with an entry explaining that this had not yet been implemented.)

In the online version of the English Short Title Catalogue provided via the Research Libraries Group (RLG) Eureka interface, the wildcard/truncation symbol is a question mark and it means zero or more characters and can appear at the end of a word only. In this version of the ESTC the substitution of i for j (and vice versa) and likewise for u and v is automatic, so jests will hit iests. By default, searches in certain fields (author, title, imprint, and subject) have the truncation principle applied to them automatically (so that Dekke hits Dekker) but only when one is searching within a single field at a time. It is possible to search more than one field at a time, and indeed there is a pseudo-field called keyword that stands for the fields author word, title word, subject word, and imprint word, taken together. In the command line search feature--for the really serious user--terms can allegedly be combined using the connector ; lim meaning 'limit the hits to those that also meet the criteria that follow' (generally, a field label followed by the search item to be found in that field) but I was unable to get this to work in anything like the manner that the help system describes. In the current version of the online ESTC proximity matching is not implemented and the help system is silent about it.

The ESTC on CD-ROM is better than the online version. The help system reports that ? means one single character and that * means any number of characters, which is true, but it fails to mention that the latter can be used only at the end of a word. This is to say, the software allows only truncation and not full wildcarding. The pseudo-field keyword stands for a few other fields taken together, although the help system does not reveal which fields these are, and it muddies the waters in the usual fashion by also using keywords to mean the terms specified for searching within a selected field. The Boolean operator not is here called andnot, presumably because someone thought it a bit closer to ordinary English, although I would have thought butnot is even closer to how people speak: Laurel butnot Hardy seems better than Laurel andnot Hardy. Proximity searches are done using the operator near, which defaults to 10 words of separation, or nearx for x number of words, and this disregards the order in which the two words occur (but they must be in the same field), or by using with and withx, which requires that the two words appear in the field (and again it must be the same field) in the order specified. The product help system uses capital letters to distinguish Boolean and proximity operators (so in fact it is described as NEARx) without mentioning whether the capitalization matters in practice. It does not. If the user knows what she is about and looks in the help system's index for the word proximity she will not find it: the near and with operators are wrongly described as (and filed with the) Boolean operators.

In Early English Books Online there is a keywords pseudo-field that represents all the searchable fields and Boolean and, or, and not are provided. Proximity searches are provided by the near.x operator (default if .x unspecified is 10) and directionality by a fby.x (mnemonic for followed by) operator. The only form of wildcarding is the truncation symbol asterisk that has to occur at the end of a word and the documentation wrongly describes it as standing for one or more occurrences of any character when in fact it stands for zero or more occurrences, so that donne* hits donne. The Editions and Adaptations of Shakespeare CD-ROM is the oldest product I looked at, and its proximity searching is done by the operators within x words of (meaning forwards or backwards) and within x words before and within x words after (providing directionality). The Boolean operators and, or, and andnot have their familiar roles, and wildcards can appear anywhere in a search term. The wildcard ? means exactly one occurrence of any character and the wildcard * means zero or more occurrences of any character (thus w*o hits wo as well as who) and the help system is perfectly accurate in describing these terms. EAS has a feature specifically for the u/v and i/j substitution: you can list the alternatives inside square brackets, so lo[uv]e hits loue and love. This is an especially powerful feature, for any number of any characters (not just i, j, u, and v) may appear in the brackets, so that c[aou]p will find cap, cop, or cup. Moreover, the feature is implemented in a way familiar to those who do the programming underlying these search engines, as we shall see.

Chadwyck-Healey's flagship product Literature Online uses keywords to mean full-text searching only, and not to mean a pseudo-field that bundles several others. Boolean logic is provided by and, or, and not and proximity searching by near.x and fby.x operators with a default separation of 10 words. In wildcarding/truncation, the asterisk stands for zero or more occurrences of any character. The help system reports that if this is used in what it calls a "phrase search" (a search specifying more than one word) the symbol must go at the end of the last word, which implies that in single-word searches it can go anywhere. This is not the case: I could not get it to work (no matter which field I searched) in any but the terminal position of a word, so I reckon the writers meant that if the asterisk is used in any word in a phrase search it must appear at the end of the phrase. Thus, this is not a true wildcard but only a truncation symbol. The symbol ? is a true wildcard--it can appear anywhere in search word--and it stands for zero or one occurrences of one character. The help system states this fact correctly, although a few months ago it was stating incorrectly that the symbol stood for exactly one occurrence of any character.

Finally, the Oxford English Dictionary on CD-ROM, which I confess has a search engine that I have barely managed to scratch the surface of. The Boolean and, or and not are available, and proximity searching is provided with near and not near with selection of modifiers before, after and before or after by radio-buttons. In truncation/wildcarding the symbol ? means exactly one occurrence of any character (so m??n misses man), but the help system does not make it explicit that zero is not an option: it just records that the question mark stands for "the occurrence of any one single character". The asterisk, however, represents zero or more occurrences of any character, and the help system makes this pleonastically explicit: "any number of characters (or no character at all"). (Zero, of course, is a number.)

The foregoing is not intended as a potted review of the products and might seem like unreasonable carping about the interfaces to what are nonetheless extraordinarily powerful resources that previous generations of scholars could scarcely dream of using. Marvin Spevack, as ever, was ahead of the field in dreaming 30 years ago of a systematized data centre holding everything known about Shakespeare's works, and was equally forward thinking in remarking 3 years ago that it is not the technology but the organization of data that keeps this desideratum from us (Spevack 2002, 83). In my view, the differences between the search engines of the major products upon which future scholarship will rely are insurmountable hurdles to their full exploitation and will drive many users away. (In case it helps ameliorate the situation, I have put on my website at www.GabrielEgan.com/whatwhere a table summarizing the above search-engine conventions, telling the reader what goes where in each product.) Two recent examples of scholarship from heavy users of electronic products in Shakespeare studies will illustrate the difficulties that even experts can get into.

MacDonald P. Jackson has been working on the authorship of Shakespeare's Titus Andronicus with George Peele as the other potential hand, revealed by his avoidance of the indefinite article an (Jackson 1998). There are five extant Peele plays and these all avoid an to the extent that only 5% of all occurrences of a or an are an whereas in Shakespeare the corresponding figure is 10%. Jackson explained that such a low figure can occur because a writer simply avoids the indefinite article altogether before a vowel by using a different construction and/or syntax. Jackson provided a table showing the ratio of a to an in a number of Shakespeare plays and Peele plays as witnessed in the electronic texts provided by the LION database. Using the figures for individual scenes in Spevack's Concordances, Jackson observed that in Titus Andronicus 1.1, 2.1, 2.2, and 4.1 (the scenes Jackson has elsewhere argued might be Peele's) an is used for about 3% of the indefinite articles, close to Peele's norm, whereas for the rest of the play the figure is 10%, which result buttresses the arguments made elsewhere that these scenes are Peele's.

A year later Jackson retracted part of what he stated about Titus Andronicus in 1998, admitting that his figures for use of an were inflated by his ignorance that LION etexts have An as a speech prefix for Lady Anne in Richard 3 and for Antipholus in The Comedy of Errors (Jackson 1999); the corrected figures weakened his argument. Jackson warned others to beware the trap he fell into and suggested they "visit the individual contexts so as to ensure that the counts include only those items with which you are specifically concerned". With a small number of hits such manual checking is feasible, but there are considerable discoveries to be made in searches that generate too many hits for this to be done. The proper moral of the story is to devise tests that would find flaws in one's methodology; computer programmers are suited to this work because they ask themselves 'how might this variable go out of bounds?'. To help with this, it is essential that searching conventions are transparent and widely-agreed upon; we need an international standard to achieve determinacy. Jackson did not describe how he used LION, but amongst its additional terms feature there is no option to exclude speech prefixes from a search. Nor indeed is there such a feature amongst the dialogue boxes in Chadwyck-Healey's English Verse Drama and English Prose Drama CD-ROM databases, which are the bases of the LION data, but in those products one could at least enter command-line searches naming particular parts of the text to include or exclude. Thus by a command line search of an in <speaker> in English Verse Drama's etext of Richard 3 one may find the 39 An speech prefixes that Jackson should have deducted from his total.

Another veteran word-counter is Thomas Merriam, who recently claimed that the stylometry in Brian Vickers's book Shakespeare, Co-author (Vickers 2002) concerning the hands in Sir Thomas More was flawed because Vickers failed to do the proper 'negative check' (Merriam 2003). Merriam pointed out that 'negative checking' (making sure an alleged similarity between known-author-text-A and unknown-author-text-B is not simply a commonplace) using LION is frustratingly awkward because the texts are in original spelling, giving as his illustration the 14 ways that to thee could appear. In this detail, Merriam was mistaken, for the search to? fby.1 th?? would catch all of these because the wildcard character ? stands for zero or one occurrence of any character. At the time Merriam wrote this, the online documentation provided with LION was also wrong on this point, claiming that ? stands for exactly one occurrence of any character. That is not what computer programmers (to whom such things are everyday affairs) would expect the character to mean and it is not indeed what the LION database software (written by programmers) actually does with this term. The only flaw in my suggestion for Merriam's search would be that one would have to eliminate the false positive to them, but that is easily accomplished with a logical not. Merriam includes ye as a form of thee, which in fact one might want to isolate, but if not it could easily be incorporated with a logical or.

My intention here is not to gloat over Jackson and Merriam's mistakes. My point is that two highly experienced researchers with sophisticated understandings of complex technical matters were, in these instances, unable to make full use of software front-ends to the databases upon which their work was based. This is a terrible indictment of the current state of electronically-enhanced research into Shakespeare. In a discipline adjacent to ours, library studies, things are done in a more professional fashion. Early adopters of information technology--and, I might add, typically rigorous in their notions of the structure of data--the librarians have developed an international standard called Z39.50 that allows communication between heterogenous databases holding bibliographical information. With software conforming to this standard, it is possible to search the catalogues of the Library of Congress, the British Library, the University of London library (and 50 others) simultaneously, which feat of electronic engineering dwarfs anything achieved by the commercial products I have surveyed. Within Z39.50 are defined standard procedures for Boolean logic, wildcarding/truncation and proximity searching, plus a host of other things that everyone who searches bibliographical data wants to do but which each database might, under its old customized front-end, have implemented differently. This is just the sort of standardization we need for the resources we use in Shakespeare studies, for even if we do not want to search, say, LION and WSB simultaneously (and there are occasions when doing so makes sense) we would, by the development of a single standard, obviate the tedious differences that currently stand in the way of our full exploitation of these resources.

We do not need to reinvent the wheel to get standard conventions and symbols for text searching, for these have been developed by computer programmers from the foundational work of the mathematician Stephen Cole Kleene and are the subject of an ISO standard. The standard is called 'regular expressions' and although powerful it is easy to use. The basic character matching symbols are:

  . matches any one alphanumeric character, so Le.r matches Lear, Leir, and indeed Le7r
  [] matches any one of the characters inside the braces, so Le[ai]r matches Lear and Leir but nothing else
  [^] matches any one alphanumeric character not inside the braces, so Le[^eo]r matches Lear and Leir but not Leer nor Leor

To cover an indeterminate number of occurrences of the desired character(s) there are repetition operators (also known as quantifiers):

  ? matches the preceding element zero or one times, so Le.?r matches Lear, Leir, Le5r, and Ler (because zero times is allowed)
  * matches the preceding element any number of times, so Le.*r matches Lear, Leir, Le555r, Ler, and Leerdammer
  + matches the preceding element one or more times, so Le.+r matches Lear, Leir, Leerdammer but not Ler (because zero times not allowed)
  {n} matches the preceding element exactly n times, so Le.{3}r matches Letter, Leader, Le555r and all other 6-letter possibilities
  {n,} matches the preceding element n or more times, so Le{3,}r matches Letter, Leader, Le555r and Leerdammer
  {n,N} matches the preceding element at least n but no more than N times, so Le.{1,3}r matches Lear, Leir, Leeer, Lester and Le123r but not Leerdammer

Finally there are symbols to represent where in a word or in a line of text the match may occur:

^ matches at the beginning of a line, so ^Lear would find a match in the line Lear goes but not in the line Go Lear

$ matches at the end of a line, so $Lear would find a match in the line Go Lear but not in the line Lear goes

< matches at the beginning of a word, so <lear would find a match in learn but not in clear

> matches at the end of a word, so >lear would find a match in clear but not learn

Finally, the backslash character \ makes the following symbol literal rather than symbolic, so that \. can be used to find actual periods.

Different implementations of regular expressions add a few features to the above, but they need not concern us here. The above tools are more than adequate for the kinds of searching that even an advanced user of the products I have been surveying would want to do. Moreover, the power of the basic tools obviates the need for proximity operators altogether, for the characters (including punctuation and spaces) between the two or more desired terms can be defined as an additional term to be searched for. That is to say, rather than looking for A followed-by-within-5-words B one could define followed-by-within-5-words in terms of the characters, spaces and punctuation that make up the separation, because a word consists of any number of alphabetic characters (represented by [abcdefghijklmopqrstuvwxyz;,\.-]*, usually abbreviated to [a-z;,\.-]*) followed by a space, and this object can be allowed to occur 1 to 5 times using {1,5}. To judge from the abstracts submitted for this seminar, participants are actively engaged in developing electronic products for Shakespeare research and they have the power to directly affect the interfaces written for these products. There is a little effort in learning the conventions and symbols of the 'regular expressions' standard, but importantly it is a transferable skill and it gives one a powerful way to think about written language. Moreover, the programmers writing the software for new products already know the 'regular expressions' standard (every programming language includes it) and they will be glad not to have to reinvent the wheel. Users will find that there are hundreds of freely-available webpages that explain how to construct 'regular expressions', starting from the most basic kind of search. Finally, as this skills base grows, real determinacy in searching large textual corpora will be possible, because we will no longer have to wonder whether a particular search's negative result indicates something significant about the data being examined or merely indicates our failure to understand the particular searching mechanism provided by the tool at hand.

Works Cited

Brown, Sarah Annes. 2000. "Arachne's Web: Intertextual Mythography and the Renaissance Actaeon." The Renaissance Computer: Knowledge Technology in the First Age of Print. Edited by Neil Rhodes and Jonathan Sawday. London. Routledge. 120-34.

Bush, Vannevar. 1945. "As we May Think." Atlantic Monthly. 176. 101-08.

Calquist, Jonas. 2004. "Medieval Manuscripts, Hypertext and Reading: Visions of Digital Editions." Literary and Linguistic Computing. 19. 105-18.

Chadwyck-Healey: A division of ProQuest Information and Learning. 1995. Editions and Adaptations of Shakespeare Version 1.0: A CD-ROM Full-text Database of Shakespeare Editions from First Printings to 1866.

Chadwyck-Healey: A division of ProQuest Information and Learning. 2004. Literature Online Third Edition: A Full-text Subscription-only Database of English and American Literature Delivered Over the Internet from Http//lion.chadwyck.co.uk:.

Cogan, Thomas. 1584. The Hauen of Health: Chiefely Gathered for All Those That Haue a Care of Their Health. STC 5478. London. Henry Middleton for William Norton.

Corns, Thomas N. 2000. "The Early Modern Search Engine: Indices, Title Pages, Marginalia and Contents." The Renaissance Computer: Knowledge Technology in the

First Age of Print. Edited by Neil Rhodes and Jonathan Sawday. London. Routledge. 95-105.
Coryate, Thomas. 1611. Coryats Crudities. STC 5808. London. W[illam] S[tansby for the author].
Crook, Norma and Neil Rhodes. 2000. "The Daughters of Memory: Thomas Heywood's Gunaikeion and the Female Computer." The Renaissance Computer: Knowledge Technology in the First Age of Print. Edited by Neil Rhodes and Jonathan Sawday. London. Routledge. 135-48.
ESTC Editorial Offices at University of California Riverside and the British Library London. 2004. English Short Title Catalog (ESTC): A Bibliography of Books Printed in Great Britain and Its Dependencies in Any Language to 1800, Delivered Over the Internet By the Research Libraries Group (Http//www.rlg.org): from Http//eureka.thames.rlg.org/cgi-bin/zgate2.prod:.
Harner, James L. 2004. World Shakespeare Bibliography Online 1965-2004 (Version 20043.) Internet URL Http//www.shakespearebib.org:.
Howard, Jean E. and Scott Cutler Shershow, eds. 2001. Marxist Shakespeares. Accents on Shakespeare. London. Routledge.
Jackson, MacDonald P. 1998. "Indefinite Articles in Titus Andronicus, Peele, and Shakespeare." Notes and Queries. 243. 308-10.
Jackson, MacDonald P. 1999. "Titus Andronicus and Electronic Databases: A Correction and a Warning." Notes and Queries. 244. 209-10.
Landow, George P. 1992. Hypertext: The Convergence of Contemporary Critical Theory and Technology. Johns Hopkins University Press. Baltimore.
Merriam, Thomas. 2003. "Correspondences in More and Hoffman." Notes and Queries. 248. 410-14.
ProQuest/Chadwyck-Healey/University Microfilms International. 2004. Early English Books Online. Internet Http//eebo.chadwyck.com:.
Rhodes, Neil and Jonathan Sawday. 2000a. "Introduction: Paperworlds: Imagining the Renaissance Computer." The Renaissance Computer: Knowledge Technology in the First Age of Print. Edited by Neil Rhodes and Jonathan Sawday. London. Routledge. 1-17.
Rhodes, Neil and Jonathan Sawday. 2000b. The Renaissance Computer: Knowledge Technology in the First Age of Print. London. Routledge.
Simpson, John. 2004. The Oxford English Dictionary Second Edition on CD-ROM (Version 3).
Spevack, Marvin. 2002. "Shakespearecomputer.horizons@." Shakespeare Newsletter. 52. 61, 82-84,86.
Thomson Gale and the British Library. 2004. English Short Title Catalogue (ESTC) 1473-1800 on CD-ROM, 3rd Edition.
Vickers, Brian. 2002. Shakespeare, Co-author: A Historical Study of Five Collaborative Plays. Oxford. Oxford University Press.
Whitehead, Jim. 2000. "As we Do Write: Hyper-terms for Hypertext." ACM SIGWEB Newsletter: The Association for Computing Machinery Special Interest Group on Hypertext, Hypermedia and the Web. 9.2-3. 8-18.
Williams, Raymond. 1958. Culture and Society, 1780-1950. London. Chatto and Windus.
Williams, Raymond. 1976. Keywords. London. Croom Helm.
Williams, Raymond. 1977. Marxism and Literature. Oxford. Oxford University Press.