**HOOVER — Trial**

<2 figures>

**The Trials of Tokenization**

Hoover, D. L.

The process of tokenizing texts is typically out of sight and almost out of mind—often handled invisibly by the analyst’s program or R script, and rarely described, discussed, or even mentioned. For ‘big data’, even if questions did arise about the nature of the word list produced, testing is not feasible. Furthermore, tokenizer accuracy is so critically affected by the state and nature of the texts that probably no general measure of accuracy or appropriateness is possible. Finally, built-in programming functions and libraries are all too often used uncritically with little realization that their output does not conform to the assumptions or expectations of the analyst. I suggest that we should pay a little more attention to the theory and practice of tokenization.1

Consider a hypothetical case. Let’s say I want to analyze 5,000 novels, have access to the texts at HathiTrust, download 5,000 novels in plain text, and tokenize them. Below is part of a page from Elizabeth Gaskell’s *Cranford*, from HathiTrust (Gaskell, 1910 [1851], 107):



Figure 1. *Cranford*, Elizabeth Gaskell, from page 107.

A human reader would have little trouble tokenizing this passage, and it is not extremely problematic, though minor OCR problems exist (mainly spacing issues around single quotation marks / apostrophes and dashes, and the line-end hyphen). I tokenized this passage with The Intelligent Archive (2012), KWIC (Tsukamoto, 2004) WordSmith Tools (Scott, 2012), Voyant (Sinclair et al., 2012), and Stylo (Eder et al., 2014).2 Even on this short text, the five programs identify three different numbers of types and two different numbers of tokens, largely because of the handling of single quotation marks. KWIC and WordSmith produce identical lists, as do Voyant and Stylo, but neither of these match The Intelligent Archive.

Now consider Charles Chesnutt’s *The House Behind the Cedars* (1900, 13), also from HathiTrust:



Figure 2. *The House Behind the Cedars*, Charles Chesnutt, from page 13.

The dialect in this passage is challenging even for human readers, and the OCR is more problematic. For example, the printed text (judging from the PDF) had spaced contractions, which explains ‘you 're’ in the fourth line from the bottom and the space in ‘lie 's’ in the first line, where the text reads “he 's.” This classic OCR problem occurs several times in this novel. And in the last line ‘you '11’ has both a space and an erroneous number 11 for the ‘ll’ (double el), another common OCR problem. Those analyzing big data usually rely on the insignificance of random error, but these and many other kinds of error are not random, and systematic error within one text, one author, one genre, or one collection could easily lead to thousands of inaccurate word frequency counts in this hypothetical study of 5,000 texts.

The use of apostrophes in the Chesnutt passage to indicate dialect pronunciations can also severely affect tokenization. Although The Intelligent Archive, KWIC, and WordSmith Tools produce exactly the same lists for this brief passage, and Voyant has the same number of types and tokens, Voyant removes all initial (but not final) apostrophes, creating different words for eight of the 97 types. Stylo removes all numbers, all initial and final apostrophes, and many internal apostrophes, retaining them only in *ain^t, gentleman^s*, and *spen^s* (replaced with a caret). It produces six more tokens and four more types than the other programs, and many more differences in the word list. Unfortunately, in Chesnutt’s short novel, more than 650 words begin and/or end with apostrophes crucial to the identity of the word, so that the word lists produced by Voyant and Stylo are quite inaccurate. Furthermore, only KWIC and WordSmith Tools let the user choose whether apostrophes and hyphens are part of a word, and whether numbers can appear in the word list or not. Only WordSmith Tools allows the user to choose whether to allow apostrophes at the beginnings and/or ends of the word as well as internally.

Obviously, the two texts examined above cause different problems, and different tokenizers are more accurate for one than for the other. Worse yet, these problems are found even in relatively carefully edited texts like those from Project Gutenberg. Although Gutenberg’s *The House Behind the Cedars* does not have spaced contractions, and correctly has *he’s* in the first line and *you’ll* in the final line, the 29 initial and final dialect apostrophes remain problematic. The Gutenberg text also represents dashes as two hyphens without spaces, creating more problems for tokenizers. The Intelligent Archive and Stylo treat these double-hyphen dashes as breaking characters, while retaining single hyphens within compound words, but KWIC, WordSmith Tools, and Voyant treat them like single hyphens, creating compounds with double hyphens where dashes are needed. The situation is still more complex if a double-hyphen is preceded or followed by a breaking character. If this sounds esoteric, consider that this short novel contains nearly 400 double-hyphen dashes (Dickens’ *Dombey and Son* has more than 2,200). And this problem, too, is highly systematic: words vary considerably in how frequently they are preceded or followed by a dash, and 1,000 dash errors per text would produce 5,000,000 errors in our hypothetical 5,000 novels. (For a practical example of the effects of error, see Matt Jockers’ discussion of topic modeling and several ‘topics’ that arose from OCR error and metadata (Jockers, 2013, 135).

It might seem that we just need more sophisticated tokenizers, but the required level of sophistication to handle double-hyphen dashes correctly is quite high, and the problems caused by apostrophes and single quotation marks cannot be correctly solved computationally at all. In some cases, not even a human reader can tokenize with certainty; in others, a computer can solve problems a human cannot.

Let’s consider a few further tokenization questions:

He said, “That’s ’bout ‘nough, sho’.”

“That’s ‘bout’, not ‘fight’; ’nough said,” Nough said.

“John tried that ‘Nough told me to’ on me,” Bill whined.

He remarked, “John said, ‘Bout starts at nine.’”

He remarked, “John said, ‘It’s ’bout time.’”

He remarked, “John said, ‘‘Bout time.’” Can these apostrophes/single quotes be handled correctly computationally? How about the two single quotes before ‘Bout’ in the last example?

I visited the U.S.S.R. Four tokens? Seven? Is the final period part of the final token?

I visited the U.S.S.R.! Four tokens? Seven? Is the final period part of the final token?

Is that C------? Is ‘C------’ the token ‘C’ followed by a dash, or the token ‘C------’? What about ‘C—’? Or ‘C-’?

C------ is here. Same questions.

Oh d--n it! Is ‘d--n’ the tokens ‘d’ and ‘n’ separated by a dash, or the token ‘d--n’? How about ‘d---n’? or ‘d-n’? or ‘G-d’?

I said--never mind. If ‘d--n’ is a token, can we prevent ‘said--never’ from being a token here?

That’s what I--a mistake, sorry. How do we get ‘d--n’ correct without failing here?

You’re a real %#@$! Three tokens? Four? Does the last include the final ‘!’? What if there were a period after the ‘!’?

You’re a real %#@$!. How about now?

I am working on a Python tokenizer that can handle most of these issues correctly, and some of these problems are fairly rare, but I despair of the possibility of creating a word frequency list that is ‘correct’ even in my own opinion. For many years I have ‘corrected’ the texts before tokenizing, but that is not a practical solution for 5,000 novels and presents its own problems.

Perhaps in sufficiently big data, the error introduced by tokenizers will not significantly alter the results, and Maciej Eder (2013) has recently shown that some corpora are remarkably resistant to some kinds of intentionally introduced error. And improving the quality of the corpus had a relatively small effect on the attribution of the Federalist Papers (Levitan and Argamon, 2006). More study seems needed before we can be complacent, however, even in large-scale problems involving only authorship or classification. For smaller-scale stylistic studies, tokenization decisions can clearly have serious repercussions. Consider Ramsay’s (2011) analysis of *The Waves*, where decisions about tokenization significantly alter the lists of men-only and women-only words and words that characterize the six narrative voices (see Hoover [2014a] and Plasek and Hoover [2014], for discussion). Another example that replicates an experience I have had several times is that a Full Spectrum analysis (Hoover, 2014b), based on Craig’s version of Burrows’s Zeta (Burrows, 2007; Craig and Kinney, 2010) can give strange results if uncorrected texts are inadvertently included. For example, in a test of Charlotte Brontë versus Anne and Emily Brontë, 11 of the 100 most distinctive words were words with inappropriate initial “apostrophes” because the novels of Anne and Emily in the analysis both used single quotation marks for dialogue.

Far from being an insignificant tool that can be taken for granted, a tokenizer expresses its author’s theory of text and can significantly affect the results of many kinds of text analysis.

**Notes**

1. As a reviewer of this paper has pointed out, the problems of tokenization have been more widely recognized recently in the NLP community. For example, Dridan and Oepen (2012) and Chiarcos et al. (2012) address and suggest partial solutions for some of the problems discussed here. Even if the problems had all been solved within the NLP community (a fact not in evidence), however, this would not diminish the force of my argument for the DH community, where there has been much less attention paid to them.

2. These programs represent a variety of those used in DH work (in order): a mature Java program with a database function, a venerable corpus linguistics program with lots of functions and user-options, a highly customizable and powerful commercial program from OUP, a widely used online tool, and a recently developed set of tools written in the currently popular R.

**References**

**Burrows, J. F.** (2007). All the Way Through: Testing for Authorship in Different Frequency Strata. *LLC,* **22**(1): 27–47.

**Chesnutt, C. W.** (1900). *The House Behind the Cedars*. Houghton Mifflin, Boston, http://babel.hathitrust.org/cgi/pt?view=plaintext;size=100;id=nc01.ark%3A%2F13960%2Ft7cr7221k;page=root;seq=25;num=13.

**Chiarcos, C., Ritz, J. and Stede, M.** (2012). By All These Lovely Tokens . . . : Merging Conflicting Tokenizations. *Language Resources and Evaluation,* **46**(1): 53–74.

**Craig, H. and Kinney, A. F.** (2010). *Shakespeare, Computers, and the Mystery of Authorship*. Cambridge University Press, Cambridge.

**Dridan, R., and Oepen, S.** (2012). Tokenization: Returning to a Long Solved Problem: A Survey, Contrastive Experiment, Recommendations, and Toolkit. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics,* pp. 378–82.

**Eder, M.** (2013). Mind Your Corpus: Systematic Errors in Authorship Attribution. *LLC,* **28**(4): 603–14.

**Eder, M., Rybicki, J. and Kestemont, M.** (2014). Stylo.

**Gaskell, E.** (1910 [1851]). *Cranford*. Houghton Mifflin, Boston, http://babel.hathitrust.org/cgi/pt?q1=twelve;id=hvd.32044097042071;view=plaintext;start=1;sz=10;page=root;size=100;seq=143;num=107.

**Hoover, D. L.** (2014a). Making Waves: Algorithmic Criticism Revisited. *DH2014*,Lausanne, Switzerland: EPFL-UNIL, pp. 202–4.

**Hoover, D. L.** (2014b). The Full-Spectrum Text-Analysis Spreadsheet. *Digital Humanities 2013*, Center for Digital Research in the Humanities, Lincoln, NE, University of Nebraska, pp. 226–29.

**The Intelligent Archive.** (2012). Centre for Literary and Linguistic Computing, University of Newcastle, Australia.

**Jockers, M. L.** (2013). *Macroanalysis: Digital Methods and Literary History*. University of Illinois Press, Urbana-Champaign.

**Levitan, S. and Argamon, S.** (2006). Fixing the Federalist: Correcting Results and Evaluating Editions for Automated Attribution. *Digital Humanities 2006*. Paris: Centre de Recherche Cultures Anglophones et Technologies de l’Information, pp. 323–26.

**Plasek, A. and Hoover, D. L.** (2014). Starting the Conversation: Literary Studies, Algorithmic Opacity, and Computer-Assisted Literary Insight. *DH2014*,Lausanne: EPFL-UNIL, pp. 305–6.

**Ramsay, S.** (2011). *Reading Machines: Toward an Algorithmic Criticism*. University of Illinois Press, Urbana.

**Scott, M.** (2012). WordSmith Tools version 6. Liverpool: Lexical Analysis Software.

**Sinclair, S., Rockwell, G. and the Voyant Tools Team**. (2012). Voyant Tools (web application).

**Tsukamoto, S.** (2004). KWIC Concordance for Windows version 4.7.