Theory-to-Query: Developing a Corpus-Analysis Method Using Computer Programming and Human Analysis

By Cana Uluak Itchuaqiyaq, Nupoor Ranade, and Rebecca Walton

ABSTRACT

Purpose: This case history reports on the process of developing a method to identify, extract, and clean string citation data from a corpus of articles to assist future studies on research methods, especially those relating citation metrics to diversity and inclusion efforts in technical communication.

Method: We developed a theory-to-query method that uses a theoretical framework, computer logic, and collaborative research design to create a custom computer program to extract data from a large corpus of text. This research method uses an iterative approach involving both human and computer analysis to complete the necessary tasks.

Results: Although we successfully created a custom computer program to parse citations, both human and computer analysis were needed to effectively extract data from the corpus. The allocation of labor (human vs. computer) was driven by the limitations of the data as well as by the limitations of human and computer abilities, rather than the type of task (e.g., repetitive, requiring pattern recognition).

Conclusion: Interdisciplinary partners should use a framework to communicate effectively in their design process to better refine a project's scope, overcome unexpected limitations, and troubleshoot. Theory-to-query is a method that combines theoretical frameworks, computer logic, and collaborative research design to create custom programs that aid analysis, such as designing a program for extracting citations from a corpus of journal articles. However, even with detailed plans and clear communication, design processes require iteration and creativity as new limitations for both human and computer analysis are identified.

Keywords: Iterative design, social justice, big data, citation analysis, computer programming

Practitioner's Takeaway:

- Theory-to-query is a method that can be used for developing custom computer programs for conducting big-data research as a team.
- Iterative design processes, if backed by structured communication schemes such as a coding logic document, allow interdisciplinary teams to work together more effectively.
- Our results demonstrate that a hybrid approach using both computer and human labor for complex data extraction improves the accuracy of results, but that the assignment of tasks to either humans or computers cannot be typified.

Theory-to-Query

INTRODUCTION

A growing body of scholarship in technical and professional communication (TPC) calls for increasing our field's diversity and inclusivity (Jones, 2016; Jones, Savage, and Yu, 2014; Savage & Agboka, 2016; Walton, Moore, & Jones, 2019). Although these terms are often paired and occasionally conflated, *diversity* refers to being represented (i.e., having a seat at the table), whereas *inclusivity* refers to being welcome and valued (i.e., having a say) (Ahmed, 2012). Arguably, one of the most influential and important ways to "have a say" in academia is to publish scholarship that is widely read and cited. Thus, analyzing TPC scholarship for citation patterns offers one way of gauging inclusivity at a field-wide level: Whose work influences scholarly conversations, to what extent, and in what ways?

Citation patterns—who gets cited and how—have implications for what our field values. For example, string citations, a type of citation in which more than one reference is cited at once, may signal insufficient engagement with particular scholarship or even marginalization (Delgado, 1992). This is because string citations involve citing a publication only in a "string" alongside other work and not discussing the publication's specific arguments or contributions. As researchers committed to improving the field's diversity and inclusivity, we sought to contribute an understanding of the field's citation patterns.

Though the field of evaluative bibliometrics is quite advanced, current bibliometric methods focus primarily on citations in general and not specifically on parsing *string* citations for analysis. Citational practices in scientific and technical fields are significantly different from the humanities. Engineering fields use numbered citation styles which, as Dowdey (1992) points out, blur distinctions among individual authors and texts by using a range of numbers, such as "[5–8]," to refer to texts and discouraging direct quotations. We discovered that existing methods of citation analysis belonged to these fields and therefore, new methods would have to be developed to map the citation patterns of our field at a large-enough scope to suggest field-wide values and to analyze these patterns at a fine-enough level of detail to indicate the presence, absence, and/or type of marginalization. This case history reports our process for developing a method for parsing string citations from journal articles for future analysis. Specifically, we describe how we collected data and solved a problem that was a precondition for conducting a thorough citation pattern analysis: developing a method to identify and extract string citations from a corpus of articles (n=777) published in five major TPC journals from 2012 to 2019. This case history addresses the following research questions:

RQ1. How can we develop methods to identify and extract string citations from a large corpus of texts?

RQ2. What factors should govern the allocation of tasks to humans and computers?

In the following sections, we first contextualize our project with a literature review, followed by a methods section detailing the process for defining project parameters, generating the corpus, creating custom code to identify string citations, and testing the outputs of the custom code. We then present our results and reflect on the process of working together to develop a method to extract string citation data. Finally, we discuss potential future studies and synthesize takeaways in the conclusion section.

LITERATURE REVIEW

To contextualize this case history, we summarize three areas of literature. First, we discuss two articles by Richard Delgado that identify how citation patterns can both reflect and reinforce marginalization. Second, we overview citation analysis methods and findings to contextualize the development of our theory-to-query method. Finally, we contextualize the research questions addressed in this case history by defining big data and summarizing challenges with developing big data methodologies such as the method we developed to identify and extract string citations.

Publications as Sites of Marginalization

Our broader research study was inspired by Richard Delgado, a prominent critical race theory (CRT) scholar. Considered to be key CRT scholarship, Delgado's 1984 "Imperial Scholar" article made serious waves in his field. Delgado used manual citation analysis techniques to trace the identity factors (white/non-white, male/female) of cited authors' and citing authors' in a set of 20 articles (1984, p. 561). Delgado found that a set of 26 authors were writing the bulk of publications discussing civil rights: a group he dubbed the "inner circle." Although these authors wrote about the oppression of minorities, they themselves were

part of the dominant population: white men. Delgado argued that an effect of discourse written almost exclusively by white men is a perennially soft take on structures of oppression. Furthermore, Delgado found that the inner circle cited other members of the inner circle almost exclusively, even though well-qualified minority scholars had also published relevant scholarship. Delgado discussed how the overwhelming dominance of white men cited in this body of scholarship may not necessarily be an act of conscious exclusion; it might just indicate heightened interest from white male scholars. However, he claims, this domination—especially in scholarship discussing marginalization and oppression of minorities and especially when minority scholars had published relevant, but uncited, work—is revealing. Citation is a strategic choice; who is cited and how they are cited are choices reflecting the priorities and values of the author.

Nearly 10 years later, Delgado (1992) conducted a follow-up citational study analyzing publication and citation practices in both CRT and radical feminist thought to see if those practices had changed. He used manual methods to investigate the citation practices of the 26 inner-circle authors to determine if they cited "minority" scholars (for Delgado's study, this meant non-white scholars and/or female scholars) in their post-"Imperial Scholar" publications (1992, pp. 1350-52). They had. He then analyzed the citation contexts in terms of purpose and polarity to determine *why* and *how* minority scholars were cited by the inner circle and the attitude conveyed by citing authors (positive, negative, etc.) towards the cited works. Delgado found that although minority scholars were getting published and cited more, often these authors were cited solely within string citations and notes. Delgado noted that string citations could signal a lack of engagement with the cited work and, ultimately, a lack of inclusivity. He discusses how relegating authors to citation instances only within string citations allows the citing author to demonstrate familiarity "with the new work while avoiding fully accounting for it in his analysis. The approach also conveys the message that minority or feminist writing is deservedly obscure, and thus only worthy of passive mention" (p. 1359). This finding indicates that, although citation patterns may reflect greater diversity, practices such as string citations may signal a lack of inclusion by declining to legitimize knowledge produced by traditionally underrepresented scholars.

As Delgado demonstrates in his studies, citation analysis is an effective way to investigate the inclusion of multiply marginalized and underrepresented (MMU) scholars in a field's publications. In addition, Lawani and Bayer (1983) point to the many reasons that citation analysis acts as an effective method to measure scholar success:

> Despite the ambiguities of citation practices, the difficulties of ascertaining why a paper is cited or not cited and the potential malpractices in citing, considerable evidence has been accumulated to suggest that citations do indeed provide an objective measure of what is variously termed 'productivity', 'significance', 'quality', 'utility', 'influence', 'effectiveness', or 'impact' of scientists and their scholarly products. (p. 61)

Because citation analysis is an effective way to measure scholar success, it is also an effective way to gauge potential marginalization.

Citation Analysis Methods to Uncover Marginalization

Campbell (2000) has argued that very few publications document the status of research methods specializing in business and technical communication. She agrees that the lack of documentation on research methods has led to incomplete, outdated knowledge of research methods which has promoted the reinvention of the wheel rather than building on prior knowledge gained through business and technical communication research. To avoid reinventing the wheel, we studied the literature on citation analysis methods in our field before developing the method described in this case study (Campbell, 2000). We came across a few works specifically relevant to our study, which are described in this section. Delgado's 1984 and 1992 studies were conducted manually. Manual methods allowed for detailed, targeted analysis of relatively small sample sizes (n=20 articles [1984, p. 561] and n=26 authors' manuscripts citing "insurgent" authors [1992, pp. 1350-52]). Delgado's citation research contributes to a larger field of study analyzing citation and its impacts on academia and law that began decades before his initial analysis. An important change in related fields like evaluative bibliometrics and scientometrics is the increasing use of computer processing in citation analysis techniques. Contemporary citation analysis

Theory-to-Query

studies often use computer techniques to analyze large sample sizes, some in excess of 5 million articles over two large databases (Tahamtan & Bornmann, 2019), that would be impossible to analyze manually.

While citation analysis studies engage a broad range of topics, scholars have used these computational techniques to investigate issues explicitly and implicitly related to potential marginalization. Hou, Li, and Niu (2011) argue that it is through counting in-text citation instances within the body of a manuscript, rather than focusing on the references list, that determinations of a cited work's influence are most "fairly" made. They assert that analyzing reference lists alone does not provide the context necessary for understanding *how* a cited work is used: "Some references are indispensable; they directly stimulate hypotheses or provide essential methods. By contrast, some other references are cited just for background information or are incidentally mentioned" (p. 724). Zhu, Turney, Lemire, and Vellino (2015) attempted to determine how the relative influence of a cited work correlated with the number of citation instances of such work within an article. They assert that "citation frequency is a measure of this influence, but a better measure would take into account how a researcher is cited, rather than giving all citations equal weight" (p. 409). Though they do not frame their research in terms of marginalization in the same way as Delgado (1984, 1992), Hou, et al. (2011) and Zhu et al.'s (2015) findings can inform the work of scholars using counting techniques to uncover issues of marginalization. Chakravartty, Kuo, Grubbs, and McIlwain (2018) critique citation practices in communication studies and the lack of diversity of authors who are cited or whose work is deemed influential: "the existing representational disparity [of minority scholars' authorship of publications] contributes to the citation disparity: White authors will always have a greater opportunity to be cited because White scholars have a greater number of publications" (p. 260). Chang (2009) makes a similar critique, echoing that of Delgado (1984) and laying out the importance of citation patterns:

> The voices of minority scholars will not be heard if we do not have the opportunity to write. In order to write, we must have a place from which to write. If we are not cited, then we are less likely to advance in the profession. This is amplified if the leaders in our field do not cite us or engage our work. If we are denied tenure, then we are very unlikely to produce legal scholarship. And even if tenure is achieved, lack of citation by the leaders in our field can limit the possibility for advancement and influence, as it is quite probable that our influence is connected to the reputation of our institution. In short, being cited as well as who cites us matters. (p. 33)

Citation analysis studies have an importance beyond understanding the mechanisms of citation practices. They provide a useful metric to understand more nebulous concepts, such as the marginalization of minority scholars.

In general, citation analysis studies use databases of bibliometric data, specifically formatted articles, and existing citation analysis programs to conduct research on large corpora of publications. For example, Tahamtan and Bornmann (2019) surveyed citation analysis studies investigating citation behavior between 2006 and 2018 and found that large-scale studies have been accommodated by technological developments, such as "the existence of machine-readable formats of publications (XML tags)" (p. 8), "journals and publishers have made scientific papers available and downloadable in XML-formatted full texts" (p. 9), and "algorithms and other services that can be used in citation context studies" (p. 10).

While XML-based citation studies' methods provide respite from the "problematic, tedious and time consuming" (Tahamtan & Bornmann, 2019, p. 8) process of computational analysis of PDF-to-TXT formatted publication samples, not all journals or publishers provide access to XML documents. Pride and Knoth (2017) state that "unless a paper is available in a structure format, such as an XML, there is a requirement for converting the original PDF file into full text prior to analysis" (p. 3). Further, available citation analysis tools center on parsing citation instances in general rather than parsing individual string citations for analysis.

Such tools, therefore, would not facilitate the analysis we wanted to conduct: focusing on string citations in research articles published in five major TPC journals. These publications are available from their respective publishers behind a paywall and only in PDF format. Such factors required that we develop a method to both read and parse string citations from our sample of journal articles. Though this process was, as Tahamtan and Bornmann describe, "problematic,

tedious and time consuming," using a PDF-to-TXT approach that involved creating a program in Python best fit the constraints of our sample, our needs, and our programming environment. In this case history, we present the development of this method, which can be useful for overcoming the challenge of analyzing big data sets that are specific to TPC.

Big Data Methods in Corpora Analysis

A number of factors enable the development of methods such as the one reported in this case history. For example, the availability and processability of full-text archives like journals and newspapers, as well as recent developments in computational technologies, enable researchers in almost all disciplines to perform corpus analysis on large datasets (Wiedemann, 2013). However, analysis mechanisms for large data sets have been looked upon suspiciously in the humanities due to the "context-dependent interpretation and the situated-ness of researchers and their aims" (Schöch, 2013). Researchers have often struggled to differentiate between big data and large data in humanities fields. Manovich (2011) observes that large corpora are only one criterion, whereas the process used for analyzing those corpora is more crucial. Big data in the information sciences has to do with the volume, variety, and velocity of information. In the humanities, especially where corpora comprise journal articles from one field, one can argue that the variety is missing. boyd and Crawford (2012) get closest to defining big data for such studies as a cultural, technological, and scholarly phenomenon that rests on the interplay of *technology* (maximizing computation power and algorithmic accuracy to gather, analyze, link, and compare large data sets); *analysis* (identifying patterns to enable economic, social, technical, and legal claims); and *mythology* (considering relationships among truth, objectivity, and accuracy). This three-faceted definition informed our approach to developing the theory-to-query method, as all three considerations guided our work.

Finding an appropriate method to analyze big data can be challenging. Manual content-analysis methods are not designed for big data sets. If used, they have been found to produce erroneous results. And conducting manual analysis of only a sampling of a large corpus can have large margins of error. An alternative approach is using algorithms and computational analysis methods like APIs and statistical data-modeling tools. But these tools are unable to understand "latent meanings or the subtleties of human language" (Lewis, Zamith, & Hermida, 2013). Other computerized approaches to content analysis also may yield satisfactory results for only surface-level analyses, sacrificing more nuanced meanings present in the analyzed texts (Conway, 2006; Linderman, 2001; Nacos et al., 1991).

These challenges have encouraged humanities scholars to find innovative solutions to address problems of analyzing large corpora. Lewis et al. (2013) argue that often, the best approach may be a hybrid that blends human and computational methods of content analysis. Hybrid methods assign tedious, repetitive tasks to computational algorithms, leaving tasks of contextual inquiry to human coders. In this way, computational methods enhance but do not replace the work of human coders, enabling researchers to work with big data while also remaining sensitive to contextual nuance. Such a division of labor sounds ideal, but as we discovered in developing the hybrid method presented in this case history, it may not be so straightforward.

METHOD

Working together to develop and use our theory-to-query method required an organized, iterative approach discussed in detail here. A theory-to-query method combines a theoretical framework (such as Delgado's 1992 critique of marginalizing string citation practices), computer logic, and collaborative research design to create a custom computer program to augment human analysis. Such methods enable researchers to apply theoretical lenses to big data research by identifying and extracting specific data from large corpora. In our case, we created a computer program to parse string citations. As Figure 1 demonstrates, theory-to-query has four major steps: 1) defining project parameters, 2) generating the initial data, 3) creating a custom computer program, and 4) testing computer program outputs. Because this is an iterative process where new information or limitations may spark the need to revisit parts of the process, clear communication and iterative scoping are crucial.

Defining Project Parameters

For communication purposes, it was important to create a coding logic document (refer to Appendix A) to clearly define our project goals. This realization

Theory-to-Query

came early on as all authors tried to communicate about the theoretical implications of string citations, as well as the nuances of identifying and extracting in-text citations using a computer program. We quickly became confused about what exactly we were asking the computer to do. To further complicate these exchanges, all three of us have different levels of experience with computer programming. These varying levels of expertise required that we find efficient, effective, and precise ways to communicate together about our project.

For example, people who are not computer programmers who use a theory-to-query approach to develop a custom computer program should first understand that computer programming tends to operate on basic rules and true/false statements. Therefore, when conveying to a programmer what the program should do, a good strategy is to describe each process and subprocess in terms of if/then/else statements. For example, we were searching for string citations. In APA-styled manuscripts, string citations appear only within a set of parentheses with each reference separated by a semicolon: e.g., (Aarons, 2012; Black Horse, 2017). So, *if* a semicolon occurs within a parenthesis, *then* the program should extract the parentheses and all text contained therein *else* the program should ignore the text and continue onto the next parenthesis. Also useful was developing a set of terms—a project-specific glossary—to precisely refer to concepts at the core of our project (refer to Table 1): e.g., APA-style String Citation, Chicago-style String Citation, PubCite, and StringPub.

Developing a glossary (Table 1) enabled us to discuss and define the precise nature of the data

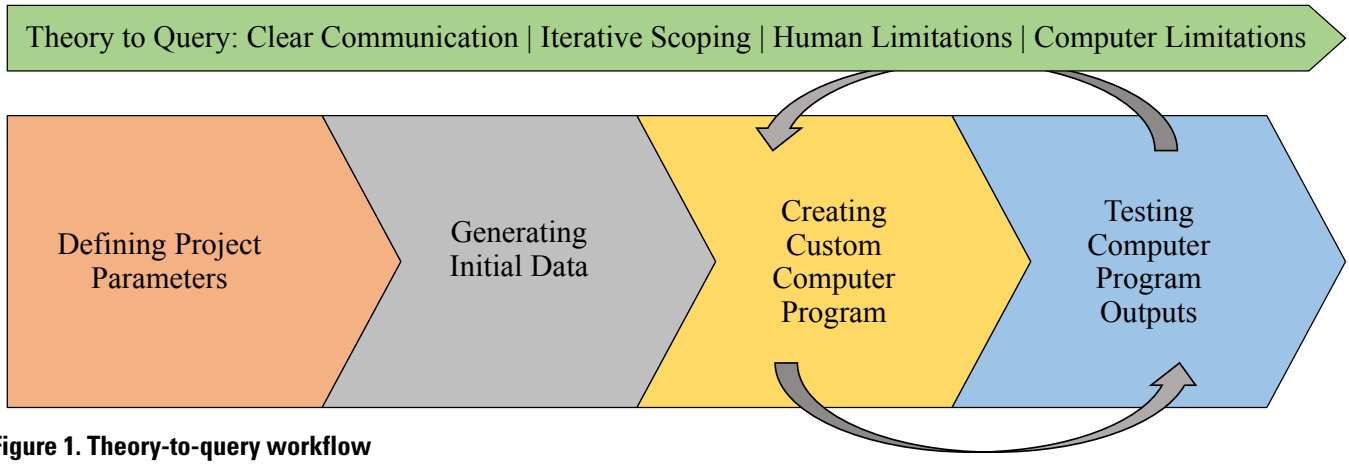

Figure 1. Theory-to-query workflow

Table 1. Project-specific glossary

Term	Definition	Example
APA-Style String Citation	An in-text reference that lists a series of two or more pieces of relevant scholarship supporting an argument. This format uses author last names and year, and lists authors in alphabetical order.	The fictitious sentence "Dogs bark at trains (Aarons, 2012; Black Horse, 2017; Change, Liu, & Billingsly, 2013; Hassan, 2015)" uses an APA-style string citation.
Chicago-Style String Citation	An in-text reference that lists a series of two or more pieces of relevant scholarship supporting an argument using a numbered list. Each number represents a particular reference, and references are listed in the order of citation within the article.	The fictitious sentences "Dogs bark at trains [1]-[4], [7];" and, "Dogs bark at trains [1-4, 7]" are Chicago-style string citations. The first example reflects IEEE's formatting, and the second example reflects JTWC's formatting.
PubCite	Any publication cited in an article within the corpus.	If one counted all the references in the reference section of an article and found that there were 45 references, the article would have 45 PubCites.
StringPub	A PubCite that occurs only as part of a string citation in any given article in the corpus.	The fictitious reference "Aarons (2012)" might occur as an in-text citation in a given article 10 different times, but if all of those citations are within strings, then Aarons (2012) is considered a StringPub.

embedded in our corpora, and creating a coding logic document enabled us to break down our major goals task by task. Identifying each necessary task was an iterative process that required revisiting the project scope and updating the coding logic document as limitations arose. For example, initially we assumed that the computer program would be able to parse individual PubCites in the references list (i.e., distinguish between each reference listed) in order to complete a series of tasks. However, because the TXT files comprising our corpora had been converted from PDF files of journal articles (which were the only available electronic format for download), much like Tahamtan and Bornmann (2019) and Pride and Knoth (2017) lament, the formatting of the TXT files created problems with parceling. For example, sometimes the file-conversion process not only introduced extra paragraph breaks within a single reference, but sometimes references were erroneously interlaced, with lines appearing out of order (refer to Figure 2). Because identifying and correcting all such problems would be enormously labor intensive, we had to reconsider all of the tasks detailed in our coding logic that required parceling information from the reference lists and come up with a new plan.

Generating Initial Data

After we had detailed the project parameters in our coding logic document, we needed to generate the initial data for our study, which consisted of a corpus of TPC journal articles. One of the first decisions was when to start: in what publication year should the corpus of TPC articles begin? Samantha Blackmon called for inclusion over 15 years ago at the CPTSC conference asking, "How do we recruit and retain minorities in our departments when there are no other minorities around?" (2004, p. 1). For years, this call went almost entirely unacknowledged in the field's scholarship until the widely acclaimed *Journal of Business and Technical Communication* special issue by Williams and Pimentel (2012) on race and ethnicity in TPC sparked more scholarship on inclusionary practices in TPC. Based on the publication date of Williams and Pimentel's milestone issue, we chose 2012 as our corpus's starting point. Our corpus consists of all articles (n=777) published between January 2012 through December 2019 from five

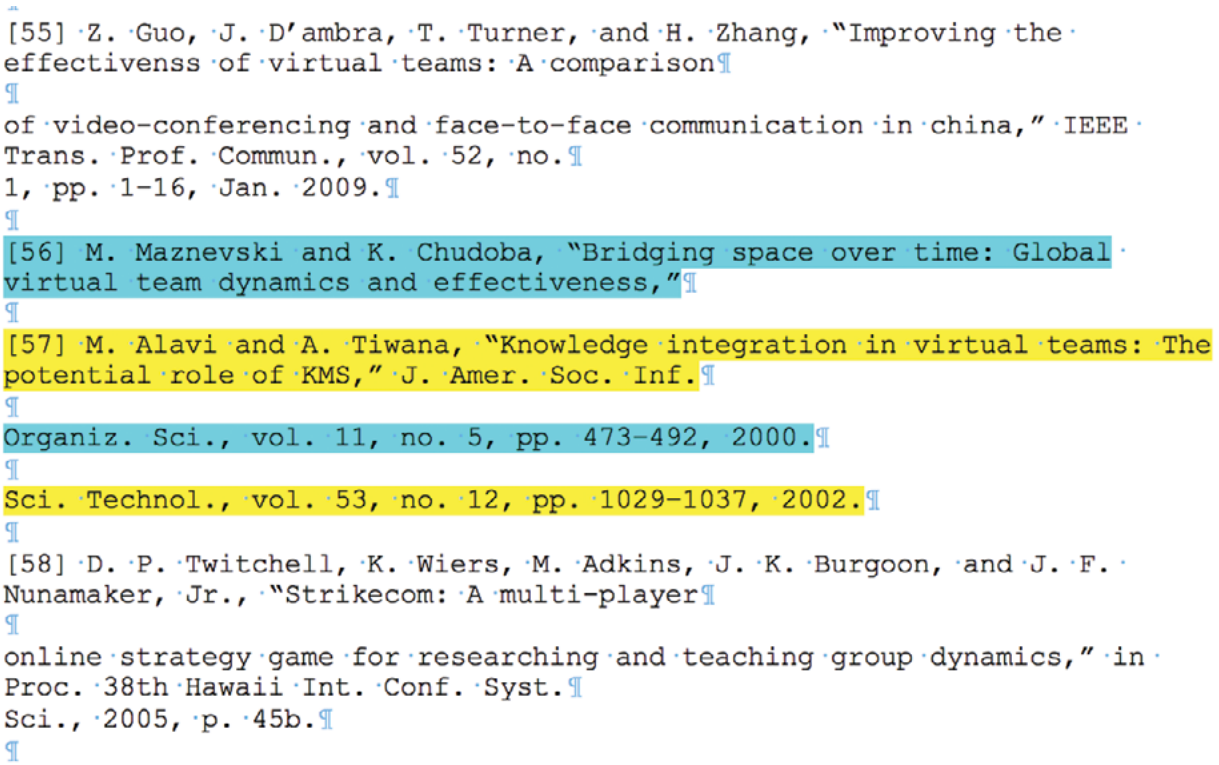

Figure 2. Image of reference section demonstrating extra paragraph breaks and ordering issues

leading TPC journals: *Technical Communication* (TC, 140 articles); *Technical Communication Quarterly* (TCQ, 166 articles); *Journal of Technical Writing and Communication* (JTWC, 158 articles); *Journal of Business and Technical Communication* (JBTC, 130 articles); and, *IEEE Transactions on Professional Communication* (IEEE, 183 articles). Each article was manually downloaded as a PDF file and then batch-converted to a TXT file. Our file-naming convention for each article indicated the year, journal name, volume and issue number, and first author last name: e.g., 2012_IEEE_v55i1_Fuller.txt.

In this case history, we focus solely on how we solved the problem of identifying and extracting string citations from a large, diversely formatted corpus. Because string citations are formatted differently in Chicago-style versus APA-style articles, we divided the initial corpus into two sub-corpora: APA-style articles (n=531) and Chicago-style articles (n=246). The Chicago-style sub-corpus contained IEEE articles and pre-2015[1] JTWC articles (n=63). The APA-style sub-corpus contained JBTC, TC, TCQ, and post-2015 JTWC articles (n=95).[1]

Creating the Custom Computer Program

We started this project using a theoretical framework (based on Delgado, 1992) to inform an initial set of research questions and then used a reflexive, iterative approach to develop a method to parse data in service of those research questions. Reflexive iteration involves visiting and revisiting data and connecting them with emerging insights, progressively leading to refined focus and understandings (Srivastava & Hopwood, 2009). Thus, the iteration in this approach to data parsing is not a repetitive mechanical task but instead a reflexive process that refines project scope and research questions (Thomas, 2006). This section describes in detail how we engaged in this reflexive, iterative process responding to both data insights and limitations of computational tools.

Text-processing tasks can be automated with either custom-built programs or readily available programs. We tested several computer applications for their capabilities to analyze the corpora, namely AntConc (Anthony, 2019), LancsBox (Brezina, Timperley, & McEnery, 2018), and Regexr (Skinner, 2020). However, each had challenges preventing their use. Initially, our goal was to develop a computer program that could identify which PubCites were StringPubs (i.e., to determine which references were cited *only* within string citations). We found that using existing applications would require individually feeding every PubCite in every article in the corpus (more than 11,000 PubCites in the 777 articles) into the computer program and manually analyzing the results. Such a process would have been impossible with such a large corpus.

We brainstormed to select an appropriate application and analytical approach, with Ranade taking the lead due to their background in computer engineering and automated data analysis. In researching the use of programming languages for corpus analysis in social sciences, we found that commercial (e.g., SAS, MATLAB) and open-source languages (e.g., R and Python) have been used to analyze data or develop applications that analyze data (McKinney, 2010). Because they can be used in conjunction with other software and have an ecosystem of ready-to-use libraries (Pedregosa et. al., 2011), languages like Python have become popular data exploratory tools in both industry and academic settings. Ranade selected Python for this project and wrote a computer program (code) to analyze the corpus. Before the corpus could be fed into the program, the corpus had to be manually pre-processed to format the articles comprising the corpus and to generate the two sub-corpora differentiated by citation style.

Ranade used Anaconda for ease of writing, editing, and demonstrating the Python code. Anaconda is an integrated development environment (IDE) which helps visualize the input and output in the same application. Visualizations supported our research team in communicating effectively. For example, the structure of the program, conditional implementations for each journal type, as well as specific operations and their correlated output can be easily demonstrated and discussed using the windows in Anaconda's layout (Figure 3).

Python has several libraries to support data analysis, and it permits the use of regular expressions to parse textual data. Regular expressions, or regexes, can be used to search files for particular patterns. We used regexes to capture and document string citations in

1 JTWC changed from Chicago-style formatting to APA-style formatting in Volume 45, Issue 2 (Spring, 2015). Volume 45, Issue 1 was formatted in Chicago style.

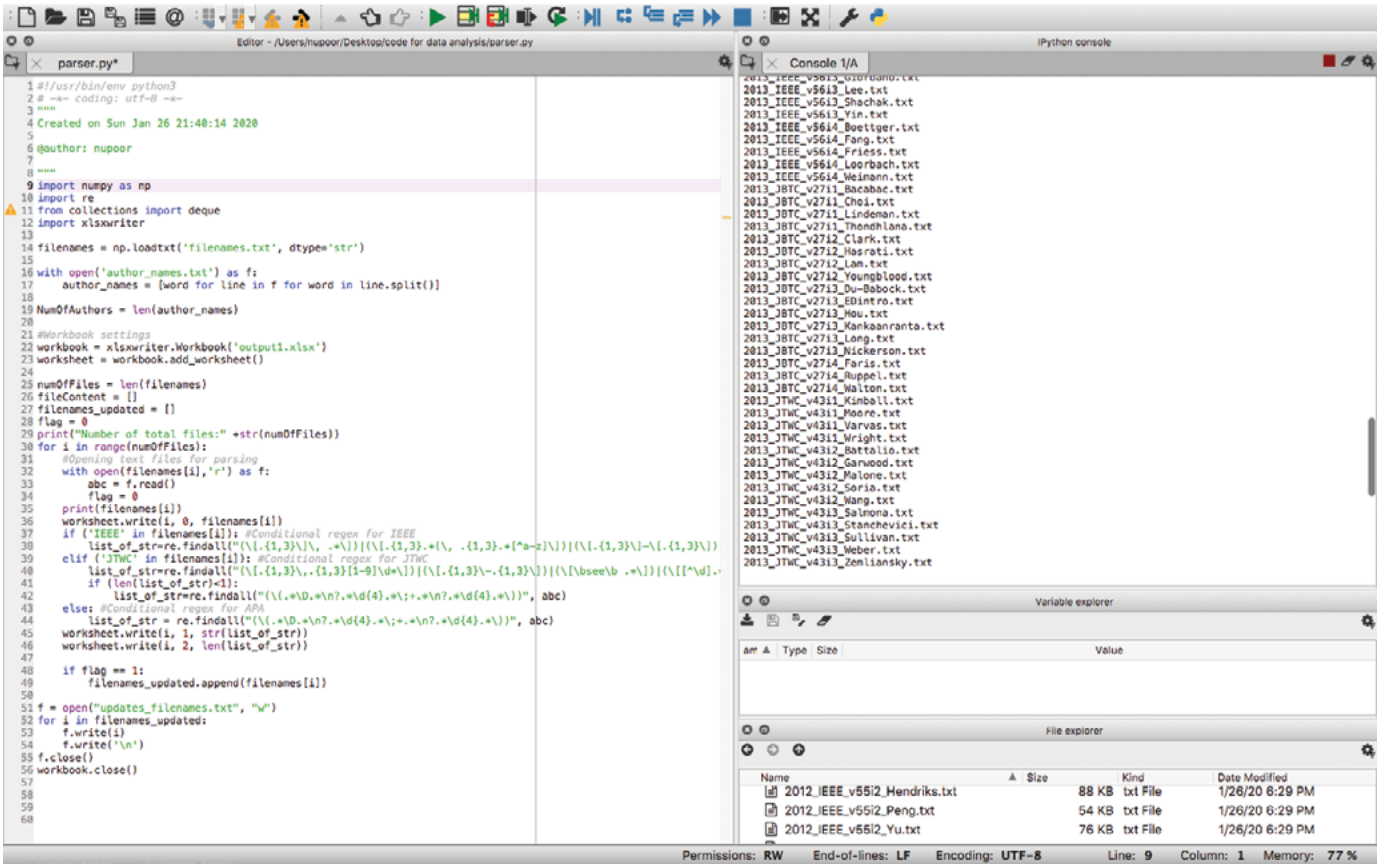

Figure 3. The split screen on the Anaconda IDE helps visualize the input and output in the same application while communicating with other researchers

both sub-corpora, but variations in citational formats required multiple regexes to be written to account for variances in citation formatting. Writing a regex is complex because regexes are highly susceptible to error. Although Python permits embedding regexes into the computer program, errors in regex formations cannot be identified through the IDE. Ranade used Regex101 to support more robust regex creation and to allow visual debugging (Ranade, 2020). We saved outputs from the code in a spreadsheet and engaged in multiple iterations of testing and code refinement to determine whether outputs were accurate.

Testing the Computer Program Outputs

While Ranade was responsible for programming, Itchuaqiyaq ensured that the results were correct and scalable to include the entire corpus. Through multiple iterations, we were able to parse the entire corpus (n=777) and get accurate results. That is, the program outputted a complete set of string citations extracted from the corpus of 777 articles, along with other information like the number of string citations in each article.

Testing was done in multiple phases. The first phase of testing found errors in file handling by the computer code and in the regexes used to identify string citations. File handling is complex. Errors can occur in converting PDF files to TXT files due to the layout, vector handling, fonts, visualizations, hyperlinks, or other variations in file formatting. We used automated batch-conversion applications to convert PDF files to TXT files. After encountering multiple errors converting particular PDF files using one such application (AntFileConverter), we used a different application (PDF to Text) to successfully convert them to TXT files. After the files were successfully converted, we discovered that the initial regex constructed to identify string citations contained errors due to the variations in citational styles used by different journals. A single regex could not be used to analyze the entire

corpus. We ended up creating three regexes to account for formatting variations and enable us to analyze the entire corpus.

The second phase of testing revealed further problems arising from regex limitations. Capable of parsing through only uniform data and finding similar patterns, regexes are created using generic patterns of the expected result. For example, if the character sequence 'dd' (used to denote two digits) is used, the program will search for only two-digit numbers. If a year is represented by 4 digits, such as 1990, it will be omitted by the parser. Therefore, to use regexes, data must be formatted consistently throughout the corpus. To identify variations that diverged from the consistent patterns identifiable by regexes, we conducted ad-hoc testing. Ad-hoc testing is a type of software testing that is performed without using test cases, plans, documentation, or systems. Usually performed by experts who developed the method, ad-hoc testing helps in identifying important defects.

For example, we suspected a potential mismatch between the patterns sought by regexes and the format of string citations in a subset of articles because the code outputs for these articles showed unusually low numbers of string citations. Itchuaqiyaq manually analyzed the articles with suspiciously low string citation counts and confirmed that the code outputs were inaccurate. Although ad-hoc testing is useful for detecting relatively large problems (e.g., problems affecting several articles in the corpus), it is less useful for identifying problems affecting a very small portion of the corpus (e.g., problems affecting only a few in-text citations in the entire corpus). To identify smaller problems, we wrote test cases and conducted iterative testing. Iterative coding and testing enabled us to develop improved regexes which resolved the inaccurate outputs. For example, we found a sizable set of JTWC articles in which the custom code identified no string citations at all. Upon manual review, we discovered that JTWC had switched citation formats from Chicago to APA, explaining why the Chicago-specific regexes failed to recognize APA-specific string citation patterns. We moved these articles into the APA sub-corpora, retested, and confirmed accurate results.

Refining the Results

We identified and extracted string citations from the two sub-corpora separately because the citation formatting differed so significantly. Chicago style's number-based citation format allowed for a hybrid

Table 2. Sample entry from string citation data spreadsheet

A: Filename	B: Total # of PubCites (references)	C: All Strings (from program)	D: # Strings (from program)	E: # Actual Strings	F: Expanded PubCites in Strings	G: Cleaned PubCites for StringPub Search (spreadsheet formula	H: # Unique PubCites in Strings (spreadsheet formula)	I: # of StringPubs	J-...: Full Citations of StringPubs
2012_IEEE_v55i1_Fuller.txt	75	[('', '[1]–[3]', '', '[1]–[3]'), ('[4], [5]', '', '', ''), **('', '[6] and Helquist, Burgoon, and Wiers [7]', ', and Wiers [7]', ''**), ('', '[12]–[17]', '', '[12]–[17]'), ('', '[15]–[17], [19]. Biros and colleagues [15]', ', [19]. Biros and colleagues [15]', ''), ('', '[22]–[24]', '', '[22]–[24]'), ('', '[25]–[27]', '', '[25]–[27]'), ('[16], [17], [19]', '', '', ''), ('', '[28]–[30]', '', '[28]–[30]'), ('', '[31]–[33]', '', '[31]–[33]'), ('[41], [42]', '', '', ''), ('[2], [3], [17]', '', '', ''), ('[12], [43]', '', '', ''), ('[14], [17], [33], [44]–[46]', '', '', ''), ('[16], [17]', '', '', ''), ('[12], [17]', '', '', ''), ('', '[28]–[30], [48]–[50]', ', [48]–[50]', ''), ('[58], [59]', '', '', ''), ('[15], [24]', '', '', ''), ('[15], [61]', '', '', ''), ('[19], [62], [63]', '', '', ''), ('[24], [68]', '', '', ''), ('[3], [40]', '', '', ''), ('[15], [24]', '', '', ''), ('[16], [64], [71]', '', '', '')]	25	24	1,2,3,1,2,3,4,5,12,13,14,15,16,17,19,22,23,24,25,26,27,16,17,19,28,29,30,31,32,33,41,42,2,3,17,12,43,14,17,33,44,45,46,16,17,12,17,28,29,30,48,49,50,58,59,15,24,15,61,19,62,63,24,68,3,40,15,24,16,64,71	1, 12, 13, 14, 15, 16, 17, 19, 2, 22, 23, 24, 25, 26, 27, 28, 29, 3, 30, 31, 32, 33, 4, 40, 41, 42, 43, 44, 45, 46, 48, 49, 5, 50, 58, 59, 61, 62, 63, 64, 68, 71	42	30	Copy & paste each StringPub reference in subsequent cells

human and computational data-refinement process, which we describe in this section. (The complexities of APA formatting required the development of a different data-refinement approach, which we will address in a future phase of this project.) Though we extracted raw data (i.e., string citations) from both sub-corpora using our custom computer program, we developed a data-refinement process for the Chicago-style sub-corpus first because its numerically based citation format made the data-refinement process much easier. The computer-extracted data from the Chicago-style sub-corpus was organized into a single spreadsheet. Table 2 presents an example of the refined, extracted data with outputs from human (manual) tasks in white cells and computer (automated) tasks in gray cells.

The raw string citation data extracted by the computer needed to be further refined for use in analysis. We conducted this refinement manually rather than computationally because of the vast amount of coding that would be required to categorize and refine this raw data. We hired a research assistant to analyze the program's output (refer to Column C and D in Table 2). Each row in the spreadsheet contains data for a single article in the Chicago-style sub-corpora. Column C contains all the string citations extracted by the custom code, with each individual string citation placed within its own set of parentheses. Column D contains a count of the string citations extracted by the program. Our research assistant read and evaluated the data in each set of parentheses in Column C and compared them against the examples of string citation formatting listed in the StringPub Identification Protocol (Appendix B) to identify errors (i.e., article text that was inaccurately extracted as a string citation). She indicated extraction errors by bolding the corresponding text in Column C. She then subtracted the errors from the total in Column D and entered the corrected total in Column E.

The research assistant then expanded all confirmed-accurate string citations from the computer program's output: for example, expanding an entry like "(', '[1]–[3]', ', '[1]–[5]')" in Column C into "1,2,3,1,2,3,4,5" in Column F. Column G uses a spreadsheet formula to convert the disorganized and repetitive entry in Column F ("1,2,3,1,2,3,4,5,") into a numerically ordered form that excludes replicates: 1, 2, 3, 4, 5. Column H uses a spreadsheet formula to count the number of PubCites listed in Column G. This count (Column H) represents the number of PubCites that are cited in a string citation in a particular journal article at least once.

To determine if a PubCite is a StringPub (i.e., if a PubCite was cited *only* as part of string citations), the research assistant used the data in Column G (Table 2) to manually check how each of the listed PubCites was cited within the journal article. This involved opening the article's file and searching for each citation (string citation or otherwise) in the article's text for every PubCite listed in Column G. The research assistant used the StringPub Identification Protocol we developed (Appendix B) to evaluate each PubCite in Column G and determine if it was a StringPub. She entered the full citation for each StringPub in the article into its own cell (Column J - end). A spreadsheet formula was used to count the number of citations pasted into the spreadsheet (Column I). While she had the article file open, the research assistant also checked the reference list (which was numbered) and entered the total number of PubCites in the article (Column B). Comparing Column I to Column B enables us to calculate the proportion of publications that are cited only in string citations compared to all cited publications for each article in the Chicago-style sub-corpus.

RESULTS

The computer program identified 24 articles within the Chicago-style sub-corpus as lacking string citations, winnowing the sub-corpus from 246 articles to 222 articles. To ensure that the computer program produced accurate results, we manually reviewed all 24 articles. We found that 11 of these articles were formatted in APA style, so we moved them to the APA-style sub-corpus. The other 13 articles were confirmed to contain no string citations; the custom code was proven accurate. The final results were as follows:

- Size of original corpus = 777 articles
- Size of Chicago-style sub-corpus (before catching the 11 APA-style articles) = 246 articles
- Size of Chicago-style sub-corpus (after removing the 11 APA-style articles) = 235 articles
- Number of articles with 1+ StringPubs in Chicago-style sub-corpus = 197
- Number of articles with 1+ string citations but no StringPubs = 25

Theory-to-Query

- Number of articles with zero string citations in Chicago-style sub-corpus = 13
- Number of PubCites in Chicago-style sub-corpus = 11,528
- Number of string citations in Chicago-style sub-corpus = 2,279
- Number of StringPubs in Chicago-style sub-corpus = 3,065

Figure 4 presents the relationship between the number of string citations and the number of PubCites in the Chicago-style sub-corpus. This figure also indicates the amount of inaccuracy in the computer-generated output versus the manually refined data.

These results were achieved with interleaving phases of manual and computational data extraction and cleaning. Although the computer program was able to parse through a large number of files and find string citations that matched regexes, sometimes those regexes inaccurately identified text as a string citation if it matched the generic pattern, even if that text was not a string citation. Our research assistant bolded those erroneous strings (Column C, Table 2) and excluded them manually cleaning the data. The inaccuracies of the custom code were eliminated through manual intervention, which required only about 6 hours to complete for the entire Chicago-style sub-corpus. To conduct the entire data-extraction process manually would have been prohibitively and enormously labor intensive. These results demonstrate that our theory-to-query method is pertinent in developing programs to extract complex information from big data sets, in this case, string citations from journal articles. Our experience with this project suggests that for big data analysis work or qualitative research for big data sets, it is important to use a hybrid approach that involves computer and human labor for the best accuracy. Our final results that combine both—initial computer outputs and human outputs (that refined and extended the computer outputs)—can be used in subsequent iterations of our program to test accuracy of extracting string citations. The computer program can be

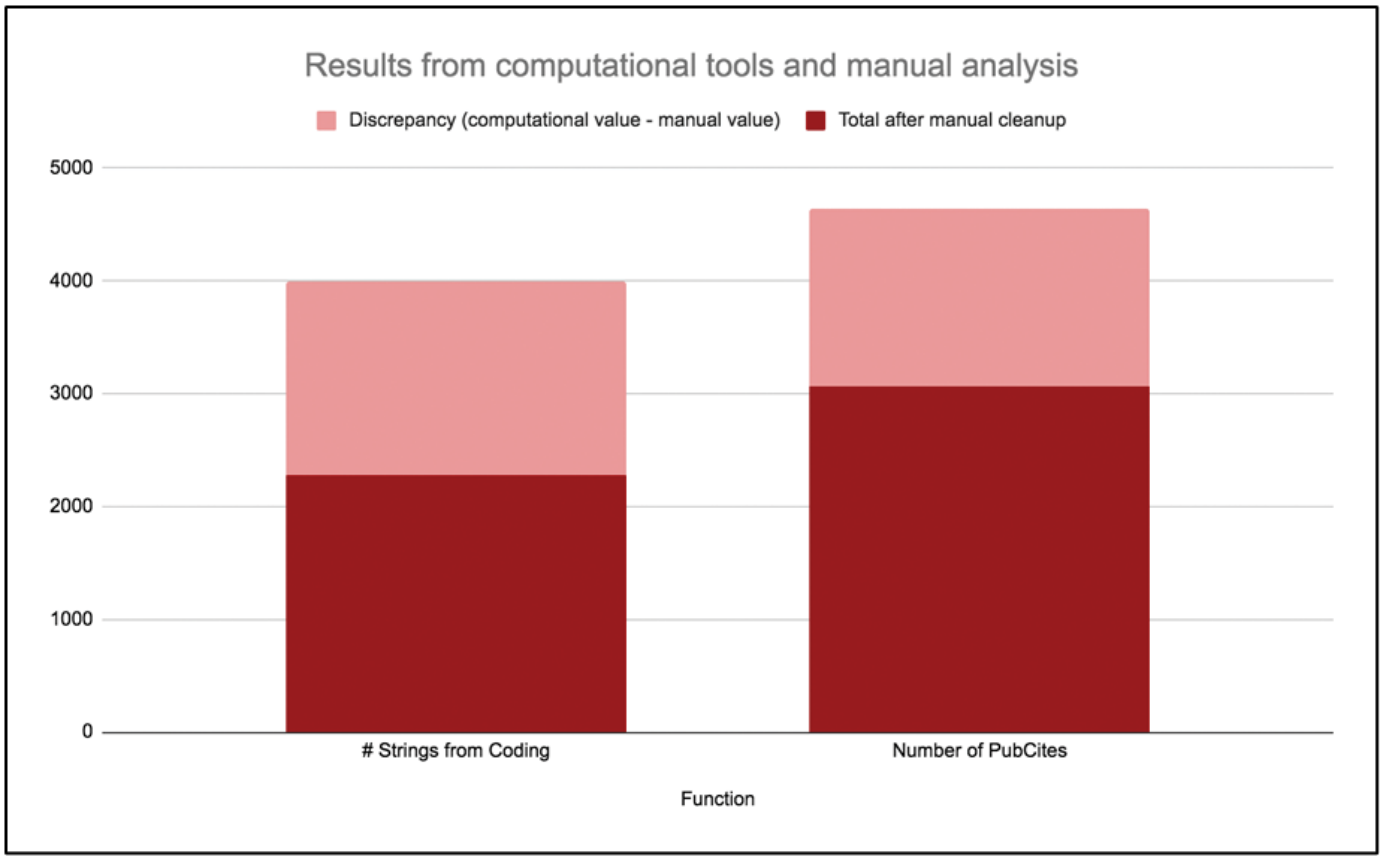

Figure 4. Amount of discrepancy resulting from the use of only computational tools. The final accurate output was generated by manual refinement

customized further for extraction of other data (such as StringPubs) on big data sets.

DISCUSSION

RQ1. Developing Methods to Identify and Extract String Citations from a Large Corpus of Texts

There are established methods of analyzing citations (in evaluative bibliometrics, scientometrics, etc.), but there are not yet computerized methods established to distinguish between types of in-text citation (e.g., identifying only StringPubs). So, in addressing our first research question, we not only developed a theory-to-query method to develop a program that could identify and extract string citations from a large corpus of texts but also identified some lessons learned that may be applicable to many big-data projects. TPC research has demonstrated that our field cares about increasing inclusivity of multiply marginalized and underrepresented scholars (Jones, 2016; Jones, Savage, & Yu, 2014; Savage & Agboka, 2016; Walton, Moore, & Jones, 2019); citational analysis is one effective method of gauging and understanding inclusivity in academic fields (Chakravartty, Kuo, Grubbs, & McIlwain, 2018; Chang, 2009). However, to our knowledge, there are no citational analysis studies published in TPC focusing on inclusivity. In designing and conducting such a study, our team identified three major challenges to citation analysis for understanding inclusivity in our field:

It is significantly more complex to analyze specific methods of in-text citation, such as string citation (per Delgado, 1992), than to analyze citations more broadly.

Python is a suitable programming language for developing programs to parse string citations, but developing custom programs in this language requires a prohibitively high level of coding skill for many in TPC.

Most TPC journals are not open source and make articles available for download only behind a paywall and only in PDF formatting. This prevents data (that is, the field's scholarship) from being available via the XML-formatted databases used in many existing citation analysis methods and tools.

Those who wish to conduct such analyses (i.e., to investigate fieldwide inclusivity through citation analysis) will find, we hope, valuable information in the description of our method, the program we developed (Ranade, 2020), and our lessons learned in developing the program through custom code. These lessons learned include the importance of developing shared language and iteratively scoping the project. Each lesson is discussed in detail below.

Clear communication: Developing shared language

In creating big-data methods that require custom computer code rather than simply using existing applications, it is important to establish and maintain clear communication, especially between researchers with different areas of expertise. Some keys to clear communication in this project are applicable to almost any collaborative research: e.g., regular team meetings in which every team member is present even if a particular meeting's agenda is relevant to only a subset of the team members' tasks. But other keys to clear communication are, we suspect, particularly relevant to big-data projects: e.g., developing a project-specific glossary that defines key terms relevant to both our process and our outcomes. Referring to such shared documentation while communicating in our team meetings saved time and supported us in pursuing shared goals.

For example, Itchuaqiyaq came to this project with a deep understanding of string citations: what they were, why they might matter, and how they differed in format from other citations. This expertise allowed her to easily distinguish the narrower category of "StringPub" (a publication cited only in string citations) from "PubCites" at large (any publication cited in any article within the corpus). But to convey that project-relevant expertise meaningfully to the other researchers, we needed the terms "PubCite" and "StringPub." Without this shared language, we found ourselves spending much of our research team meetings clarifying what we meant when discussing key aspects of the project—to include such central concerns as our research questions and specific desired outputs of the custom code. In other words, to be able to break down our major goals into specific tasks that could be coded, tested, and refined, we first needed the language to describe those tasks.

Similarly, Ranade brought to the project extensive expertise in computer programming. This expertise provided insight into which tasks might be relatively easy to perform with custom code and which tasks would not. Often the difference between the two (tasks

within reach and tasks requiring far more time and effort than we had available) was not due to the *type* of task itself (e.g., identify string citations) but to related considerations such as the ways and extents to which the data format varied. Some of these variations are meaningful to humans (e.g., Chicago style vs APA style), but other variations (e.g., a line break within a citation) are not meaningful to humans and therefore may be overlooked by researchers without coding experience. When discussing the variations for which our extraction process needed to account, we needed not only to identify all such variations but also discuss them in ways that all three researchers found meaningful if we were to collaboratively pursue a solution. In this case, that meant using examples of variations in data format to get everyone on the same page.

Our structured communication, informed by shared documentation that included a glossary and spelled out the specifics of the coding logic, may seem unnecessarily formalized. However, we found such documentation to be essential for streamlining our work and propose that such documentation could be even more valuable in industry settings, especially because the time-intensive nature of our corpus analysis method could be a barrier to practitioners. Practitioners working in industry settings may not always have access to all individuals involved in the development of their projects. For example, organizational structures in software companies may restrict technical writers to only writing roles. In these roles, technical writers may not have access to user research teams or to resources that would enable them to conduct data-based research. In the context of such restrictions, shared internal documentation could provide important context for technical writers: e.g., conveying constraints relevant to the tasks performed by colleagues in other departments/roles, as well as establishing a shared vocabulary for project tasks.

Initiating an interdisciplinary project by developing internal documentation is a practice that enacts relational values. This practice prioritizes a shared vocabulary and understanding of project goals, which can deepen mutual respect for and understanding of each member's project contributions. By first establishing a shared [written] foundation and *then* developing research methods that involve programming and humanistic skills, this practice privileges people over objects (Wolfe, 2009) and projects a humanities ethos over an engineering one (Wolfe, 2009). Therefore, we suggest that this practice is also relevant to TPC instructors who are training the future workforce. Collaborative projects in interdisciplinary courses offer many benefits, such as boosting conscious participation in problem solving through articulation and other communication practices (Ranade & Swarts, 2019), and this lesson learned about the role of communication (especially shared internal documentation) can support such projects in classroom environments.

Iterative scoping: Responding dynamically to challenges

In big-data projects requiring hybrid approaches (computer and human labor), it can be easy to lose focus on the project goals and accidentally produce outcomes that do not directly, specifically align with the research questions. We started the project with broad, exploratory questions, gradually scoping into specific research questions as the project progressed. The initial openness allowed us to prioritize extracting and cleaning data over locking in a project scope. To remind ourselves to revisit the research questions, we included them at the top of our project documents, such as the coding logic document which described specific tasks and included the glossary of shared terms. Revising project scope required revising research questions. One challenge to revising scope was distinguishing between difficult-but-possible tasks and not-possible-for-now tasks. We found team-capacity considerations useful in distinguishing these tasks. Team capacities could be expanded to some extent, but we decided that coordinating more than a three-person team would introduce too much additional complexity to an already-complex project.

As the project progressed, and we began to more fully understand the complexities and challenges of identifying and extracting string citations from such a large and diversely formatted corpus, Walton consistently drew the team back to our research questions to help us make decisions regarding a range of issues we encountered along the way. For example, as the custom code was iteratively developed and tested, we returned to the research questions to ensure that the output would, indeed, support us in answering the research questions. When differences in citation style (Chicago vs. APA) required the creation of two

sub-corpora for separate analysis, we revisited the research questions again to ensure that our process for extracting string citations would produce outputs that could be meaningfully analyzed for implications of inclusivity. Drawing upon her research experience, Walton's role in the project included ensuring that our dynamically changing approach and project scope continued to align with our research questions, paving the way for a more concrete and achievable project design. This consistent and iterative revisiting of the research questions not only clarified our decision making but also informed our understanding of the claims we could ultimately make based on the broader project.

RQ2. Assigning Computer and Human Tasks

The process for determining which tasks should be conducted manually (by human labor) versus automated (by computer code) was not as straightforward as one might assume. Common sense suggested allocating low-cognitive, repetitive tasks to computers and complex socially informed reasoning to humans. But we discovered that the messy, inconsistent, and complex formatting of data may require humans to do some of the low-cognitive, repetitive tasks. While the limitations of one (human or computer) might be the other's strength, understanding limitations of both human and computer labor is a crucial component in refining project scope. An important early task for us in designing this research project was to develop the theory-to-query method that required clearly defining our project's parameters by creating a coding logic document to act as both a programming schematic and a glossary to communicate and negotiate the limitations of this project to fit researcher capacities (both time and resources). But we suspected that even with unlimited time and resources, the data itself—its state, format, and availability—would impose limitations that direct much of who (human or computer) does what.

Limitations of human labor

While human labor might produce accurate data for repetitive tasks that require a modicum of critical reasoning, that accuracy can be impeded by the volume of those repetitive tasks. In other words, human brains grow weary and make mistakes. To help prevent such errors, we broke down the major goal of data extraction into discrete tasks that we then tested, refined, and retested. For example, expanding the computer output (Table 2, Column C) required converting ranges (e.g., [1]-[3]) into individual numbers (e.g., 1, 2, 3). This process, though simple, required the recognition and the input of several numbers that were not always listed in a string (such as the number 2) and did not always occur in order. Although all PubCites listed in any given string citation were in numerical order (e.g., [1]-[3]), each string citation in an article could have any range of PubCites (e.g., [5]-[7], [9], [43]-[52], [90]). Therefore, to mitigate human error, we used a simple spreadsheet formula to eliminate repetition and create an ordered list. Such resources (i.e., spreadsheet formulas) enable the automation of narrow, specific tasks without the development of custom tools using programming languages like Python. We see an important implication here for interdisciplinary teams in academia and in industry who are working on big-data projects: remember that automation (i.e., computer labor) can be built into tasks using a range of resources. Just because the ultimate project goal (in our case, identifying and extracting string citations) cannot be accomplished with extant tools, such tools can still prove useful to conducting or optimizing specific sub-tasks.

Limitations of computer labor

Variations in the formatting of APA-style string citations make it difficult to create a custom computer program that can identify and analyze each instance of in-text citations. Even identifying the individual PubCites listed in APA-style string citations is complicated. For example, to begin the process of determining if a PubCite is a StringPub, our custom computer program needed to first identify every string citation. Luckily, in APA style, all string citations are contained within a parenthesis with the individual PubCites separated by a semicolon. This format provided a pattern for which a computer program can search. However, when strings are identified, getting the custom program to then parcel the individual PubCites within the string proved extremely difficult. In the fictitious string citation example "(Aarons, 2012; Black Horse, 2017; Change, Liu, & Billingsly, 2013; Hassan, 2015)," the custom program would need to split this information apart and reformat it to be understood as "Aarons (2012)" "Black Horse (2017)," "Change, Liu, & Billingsly (2013)," and, "Hassan (2015)." This proved to be a challenging task due in part to

formatting problems introduced into the references list when converting TXT files into PDF files. These formatting problems made parceling out data too challenging to complete in the first stage of this project, causing us to prioritize the Chicago-style sub-corpus for data refinement, as described in the Methods section.

IMPLICATIONS AND FUTURE WORK

TPC is particularly concerned with defining and legitimizing our identity as a field (Kynell-Hunt & Savage, 2003), and one of the clearest reflections of a field's identity is its research (Rude, 2009). TPC research has been analyzed in a number of ways for its reflection of the field. For example, Rude (2009) conducted a corpus analysis of the introductions or prefaces of 109 books on TPC. Arguing that "the identity of any academic field is based in part on the research it conducts" (p. 175), Rude created a map of the field's research questions to convey that identity. Boettger and Friess pursued a similar line of inquiry, conducting a mixed-methods analysis of a large sample of TPC scholarship to identify norms in article titles (2014) and common research topics (2016). These studies demonstrate that corpus analysis offers a particularly appropriate way to understand a field as a whole, including not only its topics but also its values.

Other TPC scholarship connects the field's research and its values even more directly. For example, Blakeslee (2009) explored the field's research using a survey to identify values and constraints that shape the work of TPC researchers. Four years later, a collaborative group of TPC academic scholars and industry professionals conducted a survey to identify open research questions in the field and to compare the research priorities of scholars and practitioners (Andersen et al., 2013). These studies frame TPC research as a reflection of the field's identity, priorities, and values by tracing large-scale patterns across the field's published research and identifying implications of those patterns for what the field finds to be important and worth studying.

Arguably, analyzing a field's research for its gaps, for what and *who* is missing, offers an even more important reflection of the field's values. Some of the most influential TPC scholarship in this vein is that of Isabelle Thompson. A feminist researcher, Thompson conducted multiple corpus analyses of TPC publications using big-data techniques to investigate the field's inclusivity of women and feminism. Her 1996 review of TPC journal articles identified which theoretical frameworks, including feminist frameworks, were most influential in scholarship published between 1990 and 1994. Thompson (1999) later expanded her inquiry to include TPC articles published between 1989 and 1997: determining the frequency of scholarship that focused on women or feminism and analyzing their representation by identifying thematic trends. Arguing the importance of seeking out and valuing underrepresented perspectives, Thompson (1999) stated,

> Critiques of current and historical practice in technical communication can make us aware that common sense often means unquestioned discrimination. … In applying research to practice, we need to keep in mind the interdisciplinary roots of our discipline and the openness and flexibility that can result from this tradition. … Changing the workplace and the world—and determining the nature of technical communication as a discipline—can best be achieved by learning from all viewpoints and valuing all perspectives. (p. 175)

Thompson's work is important for understanding the field over time, gauging certain types and levels of inclusivity reflected not just in calls to action but also in our actions themselves as reflected in the body of our field's scholarship. To enable this longitudinal understanding, Smith and Thompson (2002) extended Thompson's original analysis to include scholarship published between 1997 and 2000, expanding their scope to investigate citation practices of TPC articles focusing on women and/or feminism. Their work has been extended twice more (Thompson & Smith, 2006; White, Rumsey, & Amidon, 2016). This body of scholarship demonstrates that, although the field became more inclusive of feminist topics and frameworks over the last two decades, people are still being marginalized through scholarship practices, workplace practices, and technical communication products because of their gender identity. We were inspired by this body of research, which uses big-data techniques to analyze a corpus of scholarship for what is present, who is missing, and the implications of those patterns for the inclusivity of our field.

This case history presents the theory-to-query method we developed and problem solving we

undertook to identify and extract data for a larger research project on citation patterns. Narrowing our focus to string citations offered an attainable scope that preserved our original concern with citation practices that Delgado (1992) identified as "mechanisms" of marginalization (p. 1351). In analyzing the nuances and impacts of string citations in TPC scholarship, our broader study can offer one way to gauge scholarly inclusivity at a field-wide level. To approach such a project, however, we had to first develop a method to identify and extract string citations from a large body of the field's scholarship. This initial hurdle was achievable through designing a computer program using the theory-to-query method. Future work will report findings of the broader study of citation patterns enabled by this method.

Our initial data suggests that a large proportion of scholarship is cited *only* within string citations. Table 2 provides a snapshot of this pattern: In this instance, Fuller, Marett, and Twitchell (2012) used 24 string citations in their article, and those string citations referenced 42 specific publications (42 PubCites). Of those 42 PubCites, 30 were StringPubs (i.e., cited in the article *only* within a string citation). The commonality of StringPubs (197 articles in our Chicago sub-corpus had at least 1 StringPub: 83.8%) suggests that TPC scholarship could engage more deeply and specifically with cited scholarship. We find this pattern compelling and believe it offers a rich area for future study. Delgado (1992) warns that failing to engage with a cited article's particular argument could signal marginalizing citation practices. However, we want to note that string citations, themselves, are not unethical and do not necessarily operate to marginalize. It is a matter for critical human analysis to distinguish between use of string citations that are problematic versus those that are not. With string citation data identified, extracted, and refined, we are poised to undertake that critical human analysis, thanks to the computer program we created using the theory-to-query method presented in this case history.

REFERENCES

Ahmed, S. (2012). *On being included: Racism and diversity in institutional life*. Durham, NC: Duke University Press.

Andersen, R., Benavente, S., Clark, D., Hart-Davidson, W., Rude, C., & Hackos, J. (2013). Open research questions for academics and industry professionals: results of a survey. *Communication Design Quarterly Review*, *1*(4), 42-49. https://doi.org/10.1145/2524248.2524260

Anthony, L. (2019). *AntConc (Version 3.5.8)* [Computer Software]. Tokyo, Japan: Waseda University. Retrieved from https://www.laurenceanthony.net/software

Blackmon, S. (2004). *Which came first? On minority recruitment and retention in the academy.* West Lafayette, IN: CPTSC Proceedings.

Blakeslee, A. M. (2009). The technical communication research landscape. *Journal of Business and Technical Communication, 23*(2), 129-173. https://doi.org/10.1177/1050651908328880

Boettger, R. K., & Friess, E. (2016). Academics are from Mars, practitioners are from Venus: analyzing content alignment within technical communication forums. *Technical Communication, 63*(4), 314-327.

Boettger, R.K., & Friess, E. (2014). *What are the most common title words in technical communication publications?* 2014 IEEE International Professional Communication Conference (IPCC), Professional Communication Conference (IPCC), 2014 IEEE International. doi: 10.1109/IPCC.2014.7020350

Boyd, D., & Crawford, K. (2012). Critical Questions for Big Data. *Information, Communication & Society*, *15*(5), 662-679. https://doi.org/10.1080/1369118X.2012.678878

Brezina, V., Timperley, M., & McEnery, T. (2018). *#LancsBox v. 4.x* [software]. Retrieved from http://corpora.lancs.ac.uk/lancsbox

Campbell, K. S. (2000). Research methods course work for students specializing in business and technical communication. *Journal of Business and Technical Communication, 14*(2), 223–241. https://doi.org/10.1177/105065190001400203

Chakravartty, P., Kuo, R., Grubbs, V., & McIlwain, C. (2018). #CommunicationSoWhite. *Journal of Communication*, *68*(2), 254–266. https://doi.org/10.1093/joc/jqy003

Chang, R. S. (2009). Richard Delgado and the politics of citation. *Berkeley Journal of African American Law & Policy*, *29*, 28–35. Retrieved from https://digitalcommons.law.seattleu.edu/faculty/268

Theory-to-Query

Conway, M. (2006). The subjective precision of computers: A methodological comparison with human coding in content analysis. *Journalism & Mass Communication Quarterly, 83*(1), 186-200. doi: 10.1177/107769900608300112

Delgado, R. (1984). The imperial scholar: Reflections on a review of civil rights literature. *University of Pennsylvania Law Review*, *132,* 561-578. doi: 10.2307/3311882

Delgado, R. (1992). The imperial scholar revisited: How to marginalize outsider writing, ten years later. *University of Pennsylvania Law Review, 140*, 1349-1372. doi: 10.2307/3312406

Dowdey, D. (1992). Citation and documentation across the curriculum. In M. Secor & D. Charney (Eds.), Constructing rhetorical education (pp. 330-351). Carbondale, IL: Southern Illinois University Press.

Fuller, C. M., Marett, K., & Twitchell, D. P. (2012). An examination of deception in virtual teams: Effects of deception on task performance, mutuality, and trust. *IEEE Transactions on Professional Communication*, *55*(1), 20-35. https://doi.org/10.1109/TPC.2011.2172731

Hou, W. R., Li, M., & Niu, D. K. (2011). Counting citations in texts rather than reference lists to improve the accuracy of assessing scientific contribution: Citation frequency of individual articles in other papers more fairly measures their scientific contribution than mere presence in reference lists. *BioEssays*, *33*(10), 724-727.

Jones, N. N. (2016). The technical communicator as advocate: Integrating a social justice approach in technical communication. *Journal of Technical Writing and Communication, 46*(3), 342-361. https://doi.org/10.1177/0047281616639472

Jones, N. N., Savage, G., & Yu, H. (2014). Tracking our progress. *Programmatic Perspectives, 6*(1), 132-152.

Kynell-Hunt, T., & Savage, G. J. (2003). *Power and legitimacy in technical communication: The historical and contemporary struggle for professional status* (Vol. 1). Amityvill, NY: Baywood.

Lawani, S. & Bayer, A. (1983). Validity of citation criteria for assessing the influence of scientific publications: New evidence with peer assessment. *Journal of the American Society for Information Science*, *34*(1), 59-66. https://doi.org/10.1002/asi.4630340109

Lewis, S. C., Zamith, R., & Hermida, A. (2013). Content analysis in an era of big data: a hybrid approach to computational and manual methods. *Journal of Broadcasting & Electronic Media*, *57*(1), 34-52. https://doi.org/10.1080/08838151.2012.761702

Linderman, A. (2001). *Computer content analysis and manual coding techniques: A comparative analysis.* In M. D. West (Ed.), Theory, Method, and Practice in Computer Content Analysis. Westport, CT: Ablex Pub. Corp.

Manovich, L. (2011). Trending: the promises and the challenges of big social data, in *Debates in the Digital Humanities*, ed. M. K. Gold, The University of Minnesota Press, Minneapolis, MN.

McKinney, W. G. (2010). Data Structures for Statistical Computing in Python. *In Proceedings of the 9th Python in Science Conference* (SCIPY 2010), 51-56.

Nacos, B. L., Shapiro, R. Y., Young, J. T., Fan, D. P., Kjellstrand, T., & McCaa, C. (1991). Content analysis of news reports: Comparing human coding and a computer-assisted method. *Communication*, *12*, 111-128.

Pedregosa F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*, 2825-2830.

Pride, D., & Knoth, P. (2017). Incidental or influential? A decade of using text-mining for citation function classification. *16th International Society of Scientometrics and Informetrics Conference*, Wuhan, 16-20 Oct 2017. Open Research Online.

Ranade, N. (2020). nupoor-ranade/string_citation_parser: Citations parser for string citations (APA, IEEE, Chicago) (Version v1.0). *Zenodo*. Retrieved from http://doi.org/10.5281/zenodo.3831435

Rude, C. D. (2009). Mapping the research questions in technical communication. *Journal of Business and Technical Communication, 23*(2), 174-215. https://doi.org/10.1177/1050651908329562

Savage, G., & Agboka, G. Y. (2016). Guest editors' introduction to special issue: Professional communication, social justice, and the global South. *Connexions* (4)1. http://dx.doi.org/10.21310/cnx.4.1.16

Schöch, C. (2013). Big? Smart? Clean? Messy? Data in the Humanities. *Journal of Digital Humanities*. *2*(3), 2-13.

Skinner, G. (2020) RegExr v3.7.0 [software]. Retrieved from https://regexr.com/

Smith, E. O., & Thompson, I. (2002). Feminist theory in technical communication: Making knowledge claims visible. *Journal of Business and Technical Communication, 16*(4), 441-477. https://doi.org/10.1177/105065102236526

Srivastava, P., & Hopwood, N. (2009). A Practical Iterative Framework for Qualitative Data Analysis. *International Journal of Qualitative Methods, 8*(1), 76-84. https://doi.org/10.1177/160940690900800107

Tahamtan, I., & Bornmann, L. (2019). What do citation counts measure? An updated review of studies on citations in scientific documents published between 2006 and 2018. *Scientometrics*, *121*(3), 1635-1684. https://doi.org/10.1007/s11192-019-03243-4

Thomas, D. R. (2006). A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation, 27*(2), 237-246. https://doi.org/10.1177/1098214005283748

Thompson, I. (1996). Competence and critique in technical communication: A qualitative content analysis of journal articles. *Journal of Business and Technical Communication, 10*(1), 48-80. https://doi.org/10.1177/1050651996010001003

Thompson, I. (1999). Women and feminism in technical communication: A qualitative content analysis of journal articles published in 1989 through 1997. *Journal of Business and Technical Communication, 13*(2), 154-178. https://doi.org/10.1177/1050651999013002002

Thompson, I., & Smith, E. O. (2006). Women and feminism in technical communication: An update. *Journal of Technical Writing and Communication, 36*(2), 183–199. https://doi.org/10.2190/4JUC-8RAC-73H6-N57U

Walton, R., Moore, K. R., & Jones, N. N. (2019). *Technical communication after the social justice turn: Building coalitions for action*. New York: Routledge.

White, K., Rumsey, S. K., & Amidon, S. (2016). Are we "there" yet? The treatment of gender and feminism in technical, business, and workplace writing studies. *Journal of Technical Writing and Communication, 46*(1), 27-58. https://doi.org/10.1177/0047281615600637

Wiedemann, G. (2013). Opening up to Big Data: Computer-Assisted Analysis of Textual Data in Social Sciences. *Historical Social Research, 14*, 332-358.

Williams, M. F., & Pimentel, O. (2012). Introduction: Race, ethnicity, and technical communication. *Journal of Business and Technical Communication*, *26*(3), 271-276. https://doi.org/10.1177/1050651912439535

Wolfe, J. (2009). How Technical Communication Textbooks Fail Engineering Students. *Technical Communication Quarterly*, *18*(4), 351-375. https://doi.org/10.1080/10572250903149662

Zhu, X., Turney, P., Lemire, D., & Vellino, A. (2015). Measuring academic influence: Not all citations are equal. *Journal of the Association for Information Science and Technology*, *66*(2), 408-427. https://doi.org/https://doi.org/10.1002/asi

ACKNOWLEDGEMENTS

We would like to thank our research assistant, Dylan Paisaq Tuunusaq Crosby, for her patience and valuable contribution to this project. (Quyanaqpak, paniiŋ!) We would also like to thank the Association for Teachers of Technical Writing (ATTW), whose generous Graduate Student Research Award (which Itchuaqiyaq was awarded in 2020) allowed us to pay our research assistant for her labor. We would also like to thank our colleagues John Gallagher, Scott Graham, and Ryan Omizo for their advice as we created our program.

ABOUT THE AUTHORS

Cana Uluak Itchuaqiyaq is a tribal member of the Noorvik Native Community and an assistant professor at Virginia Polytechnic Institute and State University. Cana Uluak's research addresses how mainstream academic practice often perpetuates the marginalization of underrepresented scholars and communities and consequentially interferes with diversity and inclusion efforts. She is the winner of multiple national awards, including CPTSC Bedford/St. Martin's Diversity Scholarship, the CCCC Scholars for the Dream Award, and the ATTW Graduate Researcher Award. She is available at cana@vt.edu.

Theory-to-Query

Nupoor Ranade is an Assistant Professor at George Mason University. Her research focuses on audience analysis, data analytics, user experience, and information design, primarily in the field of technical communication and artificial intelligence. Her research combines her technological background and computational skills with her ability to address knowledge gaps in the fields of technical communication pedagogy and practice. She recently received CPTSC's research grant to work in the field of data analytics for user research. She is available at nranade@gmu.edu.

Rebecca Walton is an associate professor of Technical Communication and Rhetoric at Utah State University and the editor of *Technical Communication Quarterly.* She researches how people intervene for justice in their workplaces, and her work has informed implicit bias training, policy revision, and curriculum development at multiple universities. Walton's co-authored scholarship has won multiple national awards, including three categories of the CCCC Technical and Scientific Communication Awards, as well as the Nell Ann Pickett award, and the Frank R. Smith award.

Manuscript received 8 June 2020, revised 14 September 2020; accepted 1 December 2020.

APPENDIX A: INITIAL COMPUTER CODING LOGIC SAMPLE

RQ1: At what rate is TPC scholarship citing authors only within string citations?
(i.e., What's the proportion of PubCites that are StringPubs?)

GLOSSARY:
PubCite: A single publication cited in a text.
StringPub: A publication that is cited only within string citations in a text.

LOGIC:

1. Search CORPUS (all texts, n=777) for total number of articles cited (this gives number of PubCites per text, which we could add to know the number of PubCites in CORPUS A)
    a. Spot check a few texts manually for number of articles cited
    b. Split CORPUS (all texts, n=777) into sub-corpora based on citation style.
2. If the article uses APA-style, then add to **APA-style sub-corpora**. Else, add to **Chicago-style sub-corpora**.
3. Refine search for string citations (examples in APA-style)
    Example of starting output:
    2020_TCQ_v29i1_Doodlebob.txt: (Agboka, 2014; Agboka & Matveeva, 2018; Haas, 2012; Jones, 2016a; 2016b; Ranade & Swarts, 2019; Walton Moore, & Jones, 2019, Walwema, 2018), (Agboka 2014; Haas, 2012; Jones, Savage, Yu, 2014; Walton et al., 2019)
    a. Parcel out individual pubs from strings. (This is an alphabetized list of all pubs in string citations within a single text)
    Example first refinement:
    2020_TCQ_v29i1_Doodlebob.txt: 2016b; Agboka, 2014; Agboka, 2014; Agboka & Matveeva, 2018; Haas, 2012; Haas, 2012; Jones, 2016a; Jones, Savage, Yu, 2014; Ranade & Swarts, 2019; Walton et al., 2019; Walton Moore, & Jones, 2019; Walwema, 2018
    b. Cancel string repeats (only within each text file's results). **Keep only one occurrence of each pub (PubCite) in results from each text file.**
    Example second refinement:
    2020_TCQ_v29i1_Doodlebob.txt: 2016b; Agboka, 2014; Agboka & Matveeva, 2018; Haas, 2012; Jones, 2016a; Jones, Savage, Yu, 2014; Ranade & Swarts, 2019; Walton et al., 2019; Walton Moore, & Jones, 2019; Walwema, 2018
        i. Erased extra occurrences of Agboka, 2014 and Haas, 2012.
        ii. Add the "Jones" to the "2016b" cite—I'm not sure how to do this.
            1. Do we need an index for this?
        iii. Create initial output. Make index file/s for "et al." and "multi-author" entries.
4. Remove/block search of inputs in texts after the word "REFERENCES" occurs
    a. This prevents the program from counting the reference of any particular author/text as a "citation."
5. Search text file for other non-string PubCite occurrences
    a. If PubCite in-text citation instances >=1 only **within string citations formatting**, then keep. Else, cut. (I.e., if PubCite >1 outside of string citation, then cut.)
    Example citation instance: "Haas (2012) argued that…"
    If PubCite only occurs within a string citation, even if it is multiple times, then keep.
6. Refine results to subsume Walton et al., 2019 into Walton, Moore, & Jones, 2019 results using index file created in step 3biii.
    a. This refinement should follow the "If PubCite >1 outside of string citation, then cut" rule.
    b. Individual citation could be a simple parenthetical cite (Haas, 2012), or embedded within sentences.
7. Create output spreadsheet of each text filename that has >=1 PubCite in-text citation instances >=1 only **within string citations formatting**.
8. **Save results for RQ1**.

APPENDIX B: STRINGPUB IDENTIFICATION PROTOCOL SAMPLE

StringPub Identification Protocol for APA and Chicago Documents

Definitions

A 'PubCite' is a single reference cited in a document. For example, if a document lists 45 references, it has 45 PubCites.

A 'StringPub' is a PubCite that is used only within string citation/s in a document in a "generic" fashion. For example, if a PubCite is cited 10 times in a document, but all of those citations are part of strings without any individual context, then it is a StringPub.

Identification Protocol

StringPubs are identified based on their use within a single sentence.

1. Is the PubCite located within a string citation in the document? In other words, is the PubCite located within a sentence as part of a list of *two or more* PubCites?
    a. Four examples:
        i. "Scholars are interested in balloons [1]-[5]."
            ➢ There are five PubCites in this example.
        ii. "Scholars are interested in balloons [1-5, 7]."
            ➢ There are six PubCites in this example.
        iii. "Scholars are interested in balloons (Aardvark, 2018; Beluga, 2017)."
            ➢ There are two PubCites in this example.
        iv. "Scholars are interested in balloons (Beluga, n.d., 2017, 2018, in press)."
            ➢ There are four PubCites by a single author in this example.
    b. If YES, this may be a StringPub. Proceed to the next question.
    c. If NO, this is eliminated as a StringPub. List in the REJECTS column and start the next PubCite inquiry.

2. Is the PubCite in a sentence *on its own*, or as part of *its own* short phrase or quote, anywhere in the document?
    a. Four examples:
        i. "Aardvark states, 'Scholars are interested in balloons' [1]."
            ➢ This sentence contains only one PubCite.