Research Article

A Text-Analytic Method for Identifying Text Recycling in STEM Research Reports

Ian G. Anson, *University of Maryland, Baltimore County*

Cary Moskovitz, *Duke University*

Chris M. Anson, *North Carolina State University*

Structured Abstract

- **Background:** Text recycling (hereafter TR)—the reuse of one's own textual materials from one document in a new document—is a common but hotly debated and unsettled practice in many academic disciplines, especially in the context of peer-reviewed journal articles. Although several analytic systems have been used to determine replication of text—for example, for purposes of identifying plagiarism—they do not offer an optimal way to compare documents to determine the nature and extent of TR in order to study and theorize this as a practice in different disciplines. In this article, we first describe TR as a common phenomenon in academic publishing, then explore the challenges associated with trying to study the nature and extent of TR within STEM disciplines. We then describe in detail the complex processes we used to create a system for identifying TR across large corpora of texts, and the sentence-level string-distance lexical methods used to refine and test the system (White & Joy, 2004). The purpose of creating such a system is to identify legitimate cases of TR across large corpora of academic texts in different fields of study, allowing meaningful cross-disciplinary comparisons in future analyses of published work. The findings from such investigations will extend and refine our understanding of discourse practices in academic and scientific settings.

- **Literature Review:** Text-analytic methods have been widely developed and implemented to identify reused textual materials for detecting plagiarism, and there is considerable literature on such methods. (Instead of taking up space detailing this literature, we point readers to several recent reviews: Gupta, 2016; Hiremath & Otari, 2014; and Meuschke & Gipp, 2013). Such methods include fingerprinting, term occurrence analysis, citation analysis (identifying similarity in references and citations), and stylometry (statistically comparing authors' writing styles; see Meuschke & Gipp, 2013). Although TR occurs in a wide range of situations, recent debate has focused on recycling from one published research paper to another—particularly in STEM fields (see, for example, Andreescu, 2013; Bouville, 2008; Bretag & Mahmud, 2009; Roig, 2008; Scanlon, 2007). An important step in better understanding the practice is seeing how authors actually recycle material in their published work. Standard methods for detecting plagiarism are not directly suitable for this task, as the objective is not to determine the presence or absence of reuse itself, but to study the types and patterns of reuse, including materials that are syntactically but not substantively distinct—such as "patchwriting" (Howard, 1999).

  In the present account of our efforts to create a text-analytic system for determining TR, we take a conventional alphabetic approach to text, in part because we did not aim at this stage of our project to analyze non-discursive text such as images or other media. However, although the project adheres to conventional definitions of text, with a focus on lexical replication, we also subscribe to context-sensitive approaches to text production. The results of applying the system to large corpora of published texts can potentially reveal varieties in the practice of TR as a function of different discourse communities and disciplines. Writers' decisions within what appear to be canonical genres are contingent, based on adherence to or deviation from existing rules and procedures if and when these actually exist. Our goal is to create a system for analyzing TR in groups of texts produced by the same authors in order to determine the nature and extent of TR, especially across disciplinary areas, without judgment of scholars' use of the practice.

- **Research Questions:**

  1. Is it possible to develop an algorithm for identifying cases of TR across large corpora without producing unacceptable numbers of false positives and negatives?
  2. What specific parameters of textual identification would such a system need to be programmed to identify?

- **Methodology:** Our process of creating an algorithm for identifying cases of TR across large corpora of articles generated from grants in STEM disciplines involved the following steps:

  1. Determine the type of data to be analyzed (text, symbols, visuals, etc.)
  2. Determine the level of analysis (sentence, paragraph, etc.)
  3. Determine the unit of analysis and its parameters (verbatim passages, mixed passages, string length, word quantities, etc.)
  4. Choose a method for programming the system—Latent Dirichlet Analysis, Naïve Bayes ("bag-of-words" approach), etc.
  5. Develop and refine the system through iterations of testing on sample texts
  6. Apply the system to corpora selected to represent published articles by the same author(s) emerging from specific grants.

- **Results:** After experimenting with several text analysis methods, we chose the string distance approach to create an algorithm that scores sentences on the basis of three measures of string distance similarity. Based on a set of validation checks, results demonstrate that the algorithm is a good predictor of true instances of text recycling. It does not entirely eliminate false negatives, meaning that the algorithm is best used for comparative purposes, and not as a definitive text recycling identifier.

- **Discussion:** While the sentence classifier we developed is not perfect in its identification of TR, its accuracy does not seem to be influenced by the academic subject of the text it has analyzed. This finding helps to confirm the utility of the method for comparative purposes, especially because the styles, genres, structures, and other conventions of different disciplines sometimes predict variations in discourse practices that could affect the extent to which authors engage or do not engage in TR.

- **Conclusion:** Through an analysis of existing text classification systems in the context of the specific aims of this study—eventually to identify legitimate cases of TR as part of a study of discourse practices in academic and scientific settings—it was possible through trial and error to create an algorithm that would avoid a number of confounds, including unacceptable numbers of false positives and negatives.

- **Directions for Future Research:** The decision to include a wide range of STEM fields had implications for developing our method of analysis. In particular, we needed to account for the fact that the standard structure of research reports in some social science disciplines (and even some STEM subfields) does not follow the archetypal IMRD (Introduction, Methods, Results, and Discussion) format common in much STEM research

communication. As a result, our analysis does not attempt to map quantity or characteristics of TR on the IMRD structure. Given that much of the discourse and many of the guidelines for TR in STEM view recycling in some sections (especially methods sections, but occasionally introductions as well) as being more appropriate than recycling in other parts of such papers, it would be useful to see how recycling practices align with these structures. In addition, it will be important to test the system on other kinds of corpora in addition to research articles. While we show in this paper that our algorithm has broad applicability to fields across the STEM disciplines, much work remains to uncover whether separate, genre-sensitive algorithms may be required to study TR in other contexts.

Keywords: corpus analysis, self-plagiarism, STEM writing, text classifiers, text recycling, writing analytics

1.0 Background

Text recycling (hereafter TR)—the reuse of one's own textual materials from one document in a new document—is a common but hotly debated and unsettled practice in many academic disciplines, especially in the context of peer-reviewed journal articles. Although several analytic systems have been used to determine replication of text—for example, for purposes of identifying plagiarism—they do not offer an optimal way to compare documents to determine the nature and extent of TR in order to study and theorize this as a practice in different disciplines. In this article, we first describe TR as a common phenomenon in academic publishing, then explore the challenges associated with trying to study the nature and extent of TR within STEM disciplines. We then describe in detail the complex processes we used to create a system for identifying TR across large corpora of texts, and the methods used to refine and test the system.

The purpose of this article is to detail the challenges and decisions involved in creating a system for the automated detection of recycled text across works published by the same author(s). Because it is heuristic and preliminary, it does not move beyond the testing phase to provide the results of a large-scale analysis of STEM articles, which is a work in progress. But this study does help us understand the methodological issues inherent in the study of text similarity across academic disciplines—and how theorizing about the nature of text recycling informs the choice of methodological approach.

TR is textually indistinguishable from plagiarism.[1] Both might involve the replication of phrases, sentences, or longer passages, occur with or without attribution of the source, and by definition (see Moskovitz, 2018) lack the syntax of quotation. However, while plagiarism is

[1] Note on terminology: We use the term *text recycling* (TR), rather than the historically more common "self-plagiarism," which we see as problematic. See Moskovitz, 2018.

generally condemned in the context of published, scholarly writing because it involves one author replicating another author's textual material as his or her own, without attribution, the ethics of TR (also known as "self-plagiarism") is the subject of considerable debate and uncertainty. To appreciate the ethical ambiguity of TR, consider the titles of these recent pieces:

> "Self-Plagiarism? When Re-Purposing Text May Be Ethically Justifiable" (Australasian Human Research Ethics Consultancy Services)
>
> "Self-Plagiarism: How to Define It and Why You Should Avoid It" (American Journal Experts)
>
> "'Self-Plagiarism'? You Gotta Be Kidding" (The Writing Cooperative)
>
> "Self-Plagiarism: A Misnomer" (*American Journal of Obstetrics & Gynecology*)
>
> "Self-Plagiarism: Can You Steal from Yourself?" (Texas Tech University)
>
> "Is Recycling Your Own Work Plagiarism?" (Turnitin)
>
> "Academic Self-Plagiarism: Misconduct or a Literary Art Form?" (*For Better Science*)
>
> "On Difficulty in Handling Text Recycling" (*Science Editing*)
>
> "Managing Text-Recycling: An Ongoing Issue" (*Veterinary Anaesthesia and Analgesia*)

This list shows the broad range of circulated opinions. It includes editorials in journals published by the American College of Veterinary Anesthesia and Analgesia and the Korean Council of Science Editors, Web pages from a major university (Texas Tech) and for-profit corporations (American Journal Experts and Turnitin), and blogs by both established institutions and individuals.

Whether any instance of TR is considered acceptable depends on a complex web of contextual factors including the quantity of recycled material, where it occurs, whether and how it is attributed, the nature of the source and destination texts, the discipline, and so on. Although TR occurs in a wide range of situations, recent debate has focused on recycling from one published research paper to another—particularly in STEM fields (see, for example, Andreescu, 2013; Bouville, 2008; Bretag & Mahmud, 2009; Roig, 2008; and Scanlon, 2007). An important step in better understanding the practice is seeing how authors actually recycle material in their published work. Standard methods for detecting plagiarism are not directly suitable for this task, as the objective is not to determine the presence or absence of reuse itself, but to study the types and patterns of reuse, including materials that are syntactically but not substantively distinct—such as "patchwriting" (see Howard, 1999).

Existing software for plagiarism detection, such as Turnitin, is also not suitable for large-scale analysis of text recycling. This is because these proprietary methods rely upon indexing to capture text similarity of any kind—not reuse that is directly related to an author or group of authors' prior work on a specific project. These global similarity-checking algorithms' reference

databases also do not have perfect coverage of a scholar's existing work. Our process incorporates a painstaking data collection effort which mitigates these concerns.

In the present account of our efforts to create a text-analytic system for determining TR, we take a conventional alphabetic approach to text, in part because we did not aim at this stage of our project to analyze non-discursive text such as images or other media. However, although we focus here on conventional definitions of text, we also subscribe to context-sensitive approaches to text production because we are interested in understanding varieties of TR practices as a function of different discourse communities and disciplines. Writers' decisions within what appear to be canonical genres are contingent, based on adherence to or deviation from existing rules and procedures if and when these actually exist. Our eventual goal in analyzing TR across groups of texts produced by the same authors, and to do so across corpora produced in different disciplines, is not to reach judgment about scholars' use of the practice but to determine the nature and extent of TR as a possible reflection of the values and practices of specific discourse communities.

The present work is the product of the Text Recycling Research Project (textrecycling.org), a multi-institution research initiative working to advance our understanding of TR. In this paper, our specific goal is to introduce and validate a string distance method for classifying recycled material in pairwise comparisons of a large number of research reports. This method, once validated, would allow us to eventually compare the occurrence of TR across documents, revealing over-time and cross-disciplinary patterns in its usage.

While our broader research goals include these descriptive evaluations, a method for identifying TR has broad applicability. Researchers could use a TR classifier to evaluate their own practices or to examine trends within specific fields and subfields. In addition, the approach we used to develop our classifier could be adapted for other specialized purposes. The initial testing phase of the algorithm, which we document below, also revealed many important lessons about the effective design of similarity-based text classifiers.

In the sections that follow, we present our method and assess its utility as a classifier. We introduce the string distance approach and describe how the algorithm scores sentences on the basis of three measures of string distance similarity. Based on a set of validation checks, results demonstrate that the algorithm is a good predictor of true instances of TR. It has a harder time limiting false negatives, meaning that the algorithm is best used for comparative purposes, and not as a definitive TR identifier. However, its accuracy does not seem to be influenced by the academic subject of the text it has analyzed. This helps to confirm the utility of the method for comparative purposes. In a concluding section, we describe limitations to the approach and foreshadow continued improvements to the identification of TR in the disciplines.

2.0 Literature Review

Text-analytic methods have been widely developed and implemented to identify reused textual materials for detecting plagiarism, and there is considerable literature on such methods. (Instead of taking up space detailing this literature, we point readers to several recent reviews: Gupta,

2016; Hiremath & Otari, 2014; and Meuschke & Gipp, 2013.) Such methods include fingerprinting, term occurrence analysis, citation analysis (identifying similarity in references and citations), and stylometry (statistically comparing authors' writing styles; see Meuschke & Gipp, 2013).

While there is now substantial literature on methods to detect plagiarism and the findings from such methods, few text-analytical studies have been published on authors' reuse of their own prior work. Schein and Paladugu (2001) used PubMed combined with human analysis to study the extent of "redundant" publication (often called "salami slicing"), but did not study TR per se. Sun (2013) used Turnitin to explore the effects of discipline, authorship, and language spoken on the extent of text matching with publications of one's own and others in STEM and social science fields. Sun found that authors reused material from their own prior work more often than that of others, but did not explore recycling beyond that. Horbach and Halffman (2019) used Turnitin to investigate the extent of "problematic text recycling" among authors at Dutch universities in four research areas: biochemistry and molecular biology, economics, history, and psychology. They defined "problematic" as "containing at least 10% identical passages to previously published articles," omitting from their analysis any instances of recycling that were properly cited, as well as material in methods sections. Using code they developed themselves (SplaT, or Self-Plagiarism Tool), Collberg and Kobourov (2005) conducted a small-scale, fairly informal study involving pairwise comparisons of authors' publications listed on computer science department websites from 50 schools. They did not publish their actual results, but provided a summary stating that they found a number of cases with large amounts of overlap and no citation to the source text.

More directly relevant to our work here, Roig (2005) conducted a small-scale study using a Microsoft Word macro to identify recycled material in nine review papers in a single issue of a psychology journal. Bretag and Carapiet (2007) investigated the extent of "self-plagiarism" in research publications in humanities and social sciences disciplines in Australia—which they defined as "10% or more textual re-use of any one previous publication by the author without attribution." The study used Turnitin (www.turnitin.com) to compare electronically-available publications of ten Australian authors selected in a stratified random sample. They distinguished between large-scale reuse (what we classify as *duplicate publication*) and "cut-and-paste" of smaller chunks of text. They found "self-plagiarism" in the majority of these authors' works and that most used some amount of "cut and paste" textual re-use.

Even within this limited scope of work, scholarship has focused on measuring the extent of inappropriate reuse, such as duplicate and redundant publication and uncited reuse, rather than understanding patterns of recycling more generally. Much work has been done in recent years to classify instances of plagiarism, with highly accurate methods becoming mainstream as a result (e.g., Butakov & Scherbinin, 2009; Tao, Guowei, Hu, & Baojiang, 2013). Syntax trees, program dependency graph analysis, string distance measures, and other methods have recently been applied to identify instances where plagiarizers have copied text from other sources. But in the context of TR, these developments have not been fully integrated.

Most of the aforementioned work has focused on the detection of replicated discursive material—text as conventionally defined—rather than the panoply of media that characterize nondiscursive rhetoric: film, still and animated images, sound, and the like (see Murray, 2009). Warner (2017), for example, identifies seven ways that online texts can differ from conventional print texts, including structural design, form/content relationships, and link or node strategies. Multiple media are, Warner writes, "without question the most significant allowance of the online medium that cannot be replicated in print." Such nondiscursive material is especially challenging to match without human intervention. In our study of TR, the material we analyze involves print publication of conventional journal articles available in digital form or, for purposes of machine detection, converted into digital form. These texts often include charts, graphs, and images, but we exclude these from consideration in order to create a text-matching algorithm. Further work on TR beyond this pilot project will need to find ways to scan nondiscursive media within publications of the kind, for example, currently advocated in Elsevier's "article of the future" project (https://www.elsevier.com/connect/the-article-of-the-future).

The measurement of TR in these studies is often not programmatic as a result. Because we are interested in capturing TR across a large number of texts, we require an *automated* method, like the plagiarism detection software reviewed above, that can produce comparable scores across texts. Canonical approaches to this problem have regarded text as a "unit of analysis" to be statistically classified into various categories (one of which might be "recycled," were such a category to be appropriately defined). However, given the complex patterns of TR described above, any text classifier algorithm must be developed with specific, theoretically-motivated definitions in mind. Because definitions of text recycling are not settled in the literature, we are in need of a new approach to this methodological problem. To that end, we review several types of modern text classifiers in Section 4.0, before describing the string distance method that we ultimately adopted in Section 4.3.

3.0 Research Questions

Our broader study seeks to analyze the nature and extent of TR across a range of disciplines within STEM fields. We set out to answer the following research questions:

1. Is it possible to develop an algorithm for identifying cases of TR across large corpora without producing unacceptable numbers of false positives and negatives?
2. What specific parameters of textual identification would such a system need to be programmed to identify?

4.0 Research Methodology

4.1 Determining Scope

The word *text* is commonly used as a synonym for prose—especially in distinguishing from non-prose visual materials such as graphs, tables, and photographs. However, in fields that conduct

rhetorical or textual analysis, *text* can refer either to the entirety of a work ("the text") or to any of the materials that make up the work. In our investigations of TR, including the present work, we reserve our use of the term *text* to this latter meaning—using *document* (general) or *paper* to refer to an entire work. We define TR as follows, adapted from Moskovitz (2018):

> Text recycling is the unquoted reuse of textual material from one document in a new document where (1) the material in the new document is identical to that of the source or substantively equivalent in both form and content, (2) the material serves the same rhetorical function in both documents, and (3) at least one author of the new document is also an author of the prior document. Such reuse is text recycling regardless of the presence or absence of a citation referencing the source document.[2]

The term *text recycling* immediately invokes the replication of alphabetic text—phrases or passages taken from the body of one print context and copied verbatim in another. Yet in the context of STEM research writing, TR may also include the reuse of visuals (graphs, diagrams, photographs, tables, and so on) as well as equations and other non-alphabetic symbolic material.[3] As stated previously, we chose to limit our analysis in the current work to prose for two reasons. First, recycling of prose is considerably more contentious than visuals or equations, and thus the most urgent to understand. Second, the methods required to identify and analyze visuals or equations would be both different from and more difficult than working with prose. (For example, computationally identifying diagrams that have been recycled from one paper to another would be difficult if not impossible with currently available tools.)

Given our decision to limit our analysis to prose, we chose to omit from our study those STEM fields, such as mathematics and computer science, in which entire papers frequently consist mostly of equations or other symbolic language. However, we did want to understand and compare how TR is practiced across a broad range of scientific and technical research fields. This is important for its own sake, but also for practical purposes—since knowing whether norms of TR practice are substantively and consistently different across these domains warrants different guidelines for practice. Our corpus thus includes representative disciplines in the life sciences, natural sciences, engineering, and social sciences.

4.2 Determining the Unit of Analysis

Even when limiting the scope of inquiry to prose in research reports, creating a text-analytic method for identifying instances of TR involves other methodological decisions. Most fundamental is deciding precisely what types and amounts of reused material will be identified as TR. The shorter the matched phrase, the more likely it is to be merely a common expression (*in this respect, a complete analysis, the findings will be presented, following the method described*

[2] For a discussion of citation and attribution in relation to TR, see Moskovitz, 2018.
[3] The recycling of computer code is a related but fundamentally distinct practice—in that it consists not of the means of communicating one's research findings. but of the findings themselves. We thus see the recycling of code as more akin to the reuse or repurposing of intellectual content or data than to text recycling as typically understood.

by X, etc.) or happenstance, rather than recycled text per se. Identifying such phrases as "recycled" misrepresents the occurrence, resulting in an overestimate of recycled text. And yet some instances of recycled text do occur in only a few words, such as "siphon in contact with surface." Conversely, the longer the unit of analysis, the more likely that it represents a uniquely composed string, because some specific content will be included rather than commonplace clauses that introduce or modify that content. At the extremes, we can easily decide that verbatim replication of an entire section of a published article will count as TR and three-word strings will not. But where one draws the line between these extremes is both difficult and critical, since the decision affects whether the analysis will be conservative (allowing more false negatives to reduce false positives) or comprehensive.

String length, however, is not the only consideration. Another is that STEM researchers seem to recycle text in different ways for different aims. The simplest to identify are entire verbatim sentences or longer passages. More difficult to define and to detect are passages in which recycled material is intermixed with new material. Such intermixing occurs when authors update recycled material to make it accurate for the context of the new work. Here is a hypothetical example from the methods sections of a pair of closely related studies—the first on adolescents and a follow-up study on parents:

> Source: A sample of **46 teenagers** was randomly selected from the lists of attendees for the years **2013-2017**.
>
> Destination: A sample of **32 parents** was randomly selected from the lists of attendees for the years **2014-2018**.

In this example, the text was altered only where necessary to describe the methods for the second paper. A different type of intermixing occurs when authors, believing that recycling may not be appropriate or that others may consider it inappropriate, attempt to conceal the recycled material by substituting synonyms, reordering clauses, and so on.

> Source: A sample of **46** teenagers was randomly selected from the lists of attendees *for the years 2013-2017*.
>
> Destination: **Forty-six** teenagers were selected randomly from the lists of attendees *from 2013-2017*.

Note that in this example, the textual changes do not correspond to any substantive changes in content. In other words, the new version only looks different.

Some authors recycle text by "patchwriting" (Howard, 1999)— interspersing newly-composed text with chunks of prose from their prior work. For example, the sentence "How can the system be programmed to avoid commonplaces but flag phrases that are unlikely to have appeared in any other source?" might be repurposed as follows (italics showing the identical or nearly identical words): "*How can* we *program the system* so that it doesn't identify *phrases* that probably wouldn't *appear* somewhere else?" Although the sentence is neither syntactically nor lexically identical, it says almost exactly the same thing. Because a machine is incapable of

distinguishing between strings of words that humans would either include or exclude from the analysis, without a sophisticated detection strategy, the results would be far less accurate than hand-coding.

Other decisions involve which parts of the report to include and how to handle materials that are often duplicated but wouldn't be considered recycled—including "boilerplate" text (such as lengthy names of organisms, procedures, or phenomena), references, footnotes, acknowledgements, and other periphera. Boilerplate can range from short phrases to short passages. As Roig (2015) explains, "in the sciences, the term 'boilerplate language' has been used in recent decades to describe analogous [to legal contexts] standard language usually, but not always, of a technical nature" (p. 22). More specifically, boilerplate constitutes "unique terminology and phraseology for which there are no acceptable equivalents (e.g., *Mammalian histone lysine methyltransferase, suppressor of variegation 39H1 (SUV39H1)*" (Roig, 2015, p. 24). In identifying prose as recycled, we clearly want to exclude boilerplate. Yet, from a programming perspective, there appears to be no easy way to filter boilerplate. In addition, non-specialists in the subject matter of the document may not be able to readily determine when such material is boilerplate and when it is unique to the research being reported.

Our solutions to these problems are relatively simple, given the complexity of the task at hand. We seek to develop a tool that will identify multiple kinds of TR, including patchwriting, verbatim replication, and minor substitutions. However, many extant methods are not suited to this task, as we describe below. Instead, it was necessary to develop a specialized scoring method.

4.3 Choosing a Classifier Method

Popular approaches for text classification vary in their complexity (e.g., Grimmer & Stewart, 2013; Walczak, 2017).[4] Some are supervised, meaning that human-created decision rules help the classifier to determine the category to which an unclassified text belongs. Approaches in this family include dictionary-based classifiers, which search for words and N-grams in text that increase or decrease the likelihood of a text belonging to a class (e.g., Young & Soroka, 2012). In the case of sentiment analysis, for example, specialized dictionaries are used to identify positive, negative, or neutral language.

Other simple supervised classifiers, such as Naive Bayes, compare the features of text using what is known as a "bag of words" approach (Rish et al., 2001). These models treat each word or language feature as independent, and probabilistically classify texts after being trained to recognize their occurrence in "tagged" or human-coded subsamples. However, supervised models can become increasingly complex, including "black box" models such as Support Vector Machines (SVM). These models work to perform classifications that build in more complex

[4] Across a diverse set of disciplines including English, linguistics, computer science, statistics, economics, political science, and sociology, researchers have worked to classify and categorize large repositories of text for theoretical and practical purposes.

assumptions about the conditional co-occurrence of language features. The most complex supervised text classifiers include deep neural networks and convolutional neural networks.

Other approaches, such as Latent Dirichlet Allocation (LDA), are unsupervised (e.g., Blei, Ng, & Jordan, 2003). These methods are generative in nature, meaning that they create probabilistic models of the text's data generation process without human input. The result is a set of topics that are not pre-specified, relying upon the researcher's intuition to identify which topics are meaningful in a post-hoc manner.

The many current approaches to text-as-data can draw from a wide variety of text features in order to accomplish classification. Scholars in linguistics will often rely on part-of-speech tags and other linguistic attributes in order to better discriminate texts across relevant topics. Other approaches examine the presence of characters, words, multi-word "N-grams," or sentences in texts in order to achieve precise classification.

Even more fundamentally, scholars must think carefully about what constitutes a "text" for the purposes of classification. While a restaurant rating classifier might be able to consider each online review as a separate and distinct unit, other classifiers might benefit from an approach that subdivides texts into their constituent parts, such as sections, pages, paragraphs, sentences, or even words. As with most of the design choices in automated text analysis, these decisions should be influenced by theoretical and practical considerations.

4.4 Existing Approaches

For studying TR—our interest here—the text analytical tools reviewed above are unlikely to be effective. This is because most classifiers are designed with the assumption that there exists a set of underlying theoretical classes (with corresponding sets of language features) to which we could assign new texts. For example, a classifier adept at identifying spam emails would be on the lookout for semantic features common to unwanted messages, such as product offers, requests for personal information, and surprise inheritances from obscure princes. A supervised classifier would rely upon all of the language features it has learned from a training set to flag these diverse types of spam. In the case of TR, however, each instance contains *unique* language features. Our theoretical class has no distinguishing language features other than the ones that recur in multiple texts written by the same author or authors.

Several approaches to this problem are currently in use by scholars and practitioners. Some, such as the algorithm behind Turnitin's iThenticate software, are proprietary, meaning it is difficult to ascertain the exact method used to discern text similarity. However, a wide variety of approaches used by scholarly practitioners have been detailed in the literature (e.g., Yousuf, Ahmad, & Nasrullah, 2013). Methods of "free text" plagiarism detection include lexical and semantic approaches. Lexical approaches are natural language processing (NLP) protocols that examine text at the sentence or N-gram level for similarity, often using string distance methods (White & Joy, 2004). Semantic approaches, which often require a greater amount of metadata surrounding a text, can perform very detailed analyses of the similarity of documents in which parts of speech, word order, and stylistics have been rearranged (Alzahrani, Salim, & Abraham

2011). Both approaches rely upon quantitative assessments of text similarity, often performed using string distance calculations, specialized Naïve Bayes classifiers, and Latent Dirichlet Allocation methods (Blei, Ng, & Jordan 2003).

Our present goal is nevertheless to classify recycled texts using a lexical approach. Ideally, we could do this in a binary fashion—assigning entire texts to one of the mutually exclusive categories (*recycled* and *not recycled*). However, our aim is to develop a method for studying TR rather than labeling complete texts. Given that full-length journal articles contain thousands of words, a simple binary classification of this nature would not be useful for investigating TR as a discursive practice.

Following existing literature on similarity detection, we examine texts lexically at the *sentence level* (White & Joy, 2004). By classifying sentences of journal articles into *recycled* or *not recycled* categories, we can create a continuous measure of TR at the article level and also identify patterns of TR within texts. This is the approach we have taken—calculating a TR "score" as the proportion of sentences in an article that we suspect have been recycled. These scores can then be aggregated across grants, authors, subfields, or other parameters. To accomplish this task, we developed a specialized algorithm for identifying sentence-level TR. It is not probabilistic, relying instead upon a validated score "cutoff" point to rate the occurrence of recycling in any given sentence. There are many other possible approaches to this problem, but we believe that given our aims, ours is a logical starting point.

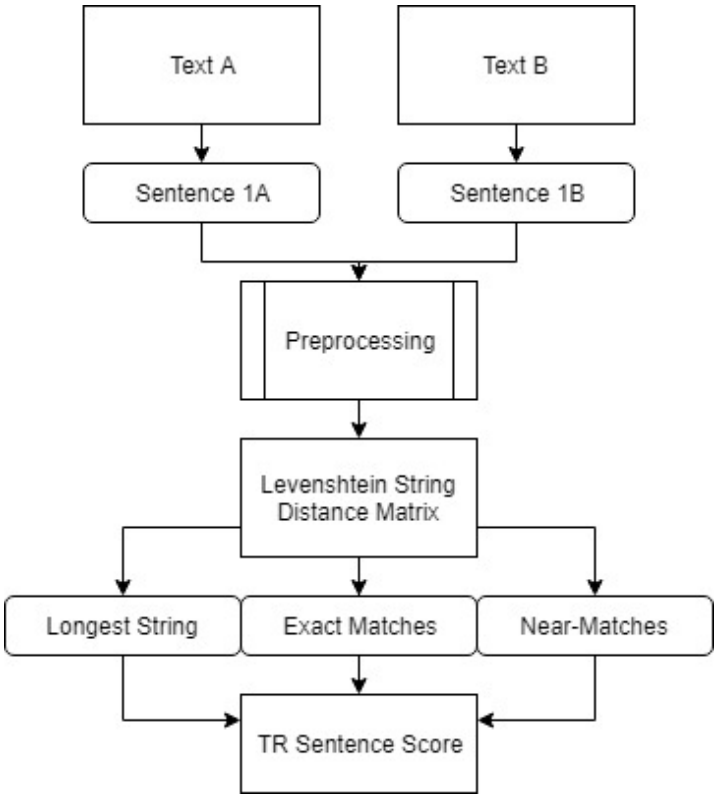

Figure 1. Diagram of TR sentence scoring process.

In 2018, we developed a classifier algorithm that relies upon the Levenshtein distance method (e.g., Su et al., 2008). As seen in Figure 1, this method compares the text features for each sentence in one document against all the sentences in another document written by the same author or research team. The classifier algorithm was designed to score articles according to a continuous measure of pairwise sentence similarity composed of three subscores: *Longest String*, *Exact Matches*, and *Near-Matches*. Below, we briefly describe this process before evaluating the effectiveness of the scoring method.

4.4 Preprocessing

The algorithm extracts a text (hereafter $T1$), and compares it to another text (hereafter $T2$). Next, it breaks both article texts into their constituent sentences using a sentence tokenizer.[5] For each sentence in the first article, $T1_i$, the algorithm performs a comprehensive comparison with all sentences in $T2$. The result is a set of $N_{T1} * N_{T2}$ total sentence comparisons.

Because we are interested in studying how authors reuse textual materials from one document in subsequent documents, our algorithm is designed with a temporal direction; it repeats for every chronologically ordered pair of texts contained in the dataset. The algorithm pairs the earliest text with all four later texts, the second-earliest text with the three published after it, and so on, until from each set of five documents in a grant we obtain 10 pairwise document comparisons.

Sentences are preprocessed in several steps. Before texts are made lowercase to facilitate comparison, we remove capitalized terms other than the first word in each sentence. While this step could theoretically reduce the capacity of the algorithm to identify TR since it strips some information from the data, we found that removing these capitalized terms helps eliminate false positives when working with STEM research reports. This is because technical scientific writing often incorporates various proper nouns, such as the names of chemical compounds, biological processes, and patented technologies. Because such terms are likely to be common to all research papers in the same, specific research area, including such terms in calculating the TR score will likely increase the rate of false positives. Removing these terms prior to sentence comparison allows us to compare the similarity of other language features that are more in line with our theoretical assumptions about TR in practice. After lowercasing, we remove numbers and symbols from the sentences, which also helps reduce false positives stemming from the use of conventional weights, measures, and other common aspects of experimental protocols.

4.5 Sentence Scoring Approach

For each cleaned sentence pair $T1_iT2_j$, we construct a rectangular *Levenshtein string distance matrix* (as seen in Figure 1). The cells of this matrix contain numerical scores that report the similarity of the words in the two sentences. The first row in the matrix compares word $T1_{i1}$ to

[5] Sentence tokenizers generally use periods and capitalization to perform sentence separation, but they are sensitive to common abbreviations and other irrelevant uses of periods.

all words $T2_{j1:N2}$; the first column compares words $T1_{i1:N1}$ to $T2_{j1}$. Matrix position [1,1] refers to a comparison between the first word of sentence i and the first word of sentence j. Within each matrix cell, the Levenshtein distance (also known as the edit distance) is calculated (Yujian & Bo, 2007; Zini, Fabbri, Moneglia, & Panunzi, 2006).[6] This value is defined as the number of insertions, deletions, and substitutions needed to transform one word into another. Cells with a value of 0 are identical word pairs, because no replacements would be required to make the two words match exactly. Cells with a value of 1 represent a word pair with only one differing letter (such as *cat* and *hat*), and cells with larger numbers represent increasingly dissimilar words.[7]

4.6 Sentence Subscore Calculation

To calculate the first subscore, *Longest String*, we measured diagonals in the word pair matrix. These diagonals contain Levenshtein scores for ordered sequences of word pairs. For example, we might consider the following pair of sentences: "The quick brown fox jumps over" and "The lazy brown dog jump under." Figure 2 shows the matrix of Levenshtein distances in pairwise comparisons for these two sentences. The diagonal of the matrix, shown here, contains the scores for each word from the first sentence against the word in the same position in the second. The other cells would contain the remaining pairwise comparisons of words from sentence 1 against words in sentence 2. For example, cell [1,2] would report the Levenshtein distance of *quick* in sentence 1 against *the* in sentence 2.

[6] Many other word-similarity scoring methods exist, such as Cosine similarity, but we find that the Levenshtein distance is helpful for the present purposes due to its simplicity and its focus on character similarity.
[7] For example, the pair *cat* and *dog* would have a Levenshtein distance of 3, because all three letters in *dog* would need to be replaced with other letters to form *cat*. *Horse* and *mouse* would be scored 2, because only the letters M and U require transformation for the words to be exact matches.

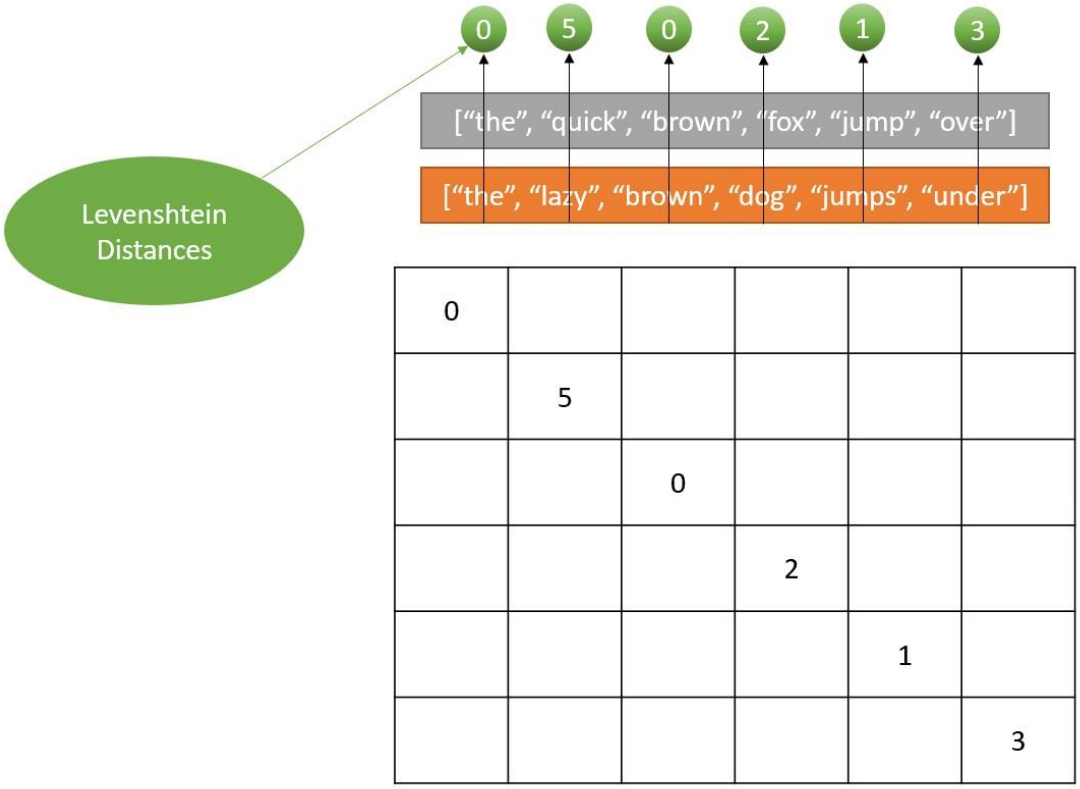

Figure 2. Calculating the diagonal of a word-pair matrix using Levenshtein distance. Sentence 1 in gray; sentence 2 in orange.

In the case of Figure 2, the longest set of consecutive 0s in the diagonal is 1, because there are no adjacent identical words. In the case of the sentences "The quick brown fox jumped over" and "The quick brown fox jumped under," however, the *Longest String* score would be 5. Note that we take the *longest* consecutive string in the matrix by evaluating multiple diagonal patterns in the case of rectangular matrices. This means that if we evaluated "I cannot believe the quick brown fox jumped over" with the latter sentence in the pair above, the algorithm would still identify the (now off-diagonal) matching pattern despite the presence of a non-matching clause at the beginning of one sentence.

Because writers often recycle text through patchwriting and word substitution, recycled passages often don't always involve long identical strings. Matrices with many values of 0 in any position indicate texts that contain identical words—though they may not be in the same order. While the *Longest String* score turned out to be the most important and informative for our purposes, as it identifies sentence chunks with many identical word features, this score was not sufficient for the accurate identification of TR in many cases.[8] We capture a second score, *Exact*

[8] Readers might also wonder whether this measure "penalizes" very short sentences. It is possible that a completely recycled sentence of only a few words will have a lower diagonal similarity score than a very long sentence with a few, scattered matching terms. Efforts to scale this measure by sentence length dramatically reduced the accuracy of

Matches, which counts all instances of 0 in the entire word pair matrix. This second measure is a bag of words approach that simply counts identical words regardless of their position in the sentence. Because long, unrecycled sentences are likely to have a number of words in common, the *Exact Matches* score is dramatically less accurate than the diagonal method when used alone. But used together, the two approaches allow us to more reliably capture two potentially common forms of TR. The former method is especially useful when writers have copied sentences from whole cloth, perhaps changing a word or two in order to disguise the replication. The latter helps us when text recyclers seek to move sentence components around in order to further avoid detection. The combination of terms still produces a high score in the bag-of-words approach.[9]

Finally, because TR often involves small adjustments in prose to accommodate the context of the new document, we add a "fuzzy" matching, bag-of-words approach that can detect those passages featuring slight changes to replicated words. This *Near-Match* score contributes to the analysis by counting words with Levenshtein distances of 1 and 2 (whereas the other two measures count only zeroes). While very short words in the matrix must be excluded from Near Match scoring since they almost always have high Levenshtein distances, this measure does help to capture some instances of TR involving minor changes.

Used together, these scores allow us to rate each sentence pair across all the texts we wish to compare. The result is our *TR sentence score*—a continuous statistic with an arbitrary scale running from zero to infinity. A score of zero on this scale indicates that there were no exactly or partially matching terms anywhere in the word pair matrix. A score of 10 on the scale could mean that there were five exactly matching words in a row, along with several other partially and fully matching terms elsewhere in the sentence. It could also indicate a long string of exactly ten matching words in a row, or some variety of exactly and partially matching words. Regardless, the TR score is not intended to systematically identify every instance of TR, and the magnitude does not represent some intrinsic characteristic of the sentence. Instead, by carefully calibrating the score by adjusting the contribution of each measure, the TR score can do the following: (1) quantify the relative amount of TR between sets of documents, (2) identify pairs of documents that do or do not contain any recycled material, and (3) reveal patterns of recycled text within documents.[10]

To further illustrate these principles, consider the following pairs of sentences, which were generated by the authors to resemble real scientific text:

---

the algorithm. Very short sentences often gave false positives, occasionally because these were sentence fragments that had been incorrectly broken into a separate sentence by the sentence stemmer. This was especially the case when it came to highly technical language, which occasionally contained features that the stemmer interpreted as a sentence break. As a result, we retain the original measure, even though it may still be sensitive to extremely long and short sentences.

[9] We also remove common stopwords from analysis in this scoring method, to avoid the false positive rates that might result from very common words being counted.

[10] Although Li and Bo (2007) normalize their scoring method, the present approach evaluates the score as is.

> "Each bacterium was grown on agar plates using commonplace streaking methods."
>
> "Bacteria were developed using agar plates using commonplace streaking methods of application."

These two sentences would likely be identified as containing text recycling. The *Longest String* score would be 6 ("agar plates using commonplace streaking methods,") and the *Exact Matches* score would be 7. We would also identify one *Near-Match*. Together, this evidence would likely produce a combined TR sentence score that rates quite highly compared to most other sentences.

The next step in the process is to determine what constitutes a score that indicates a highly-credible case of TR. In doing so, we use human-coded instances of TR to form a decision rule for identifying sentence-level TR. The aim of comparing the machine scores to human coding is to arrive at a scoring threshold for classifying a sentence as having recycled text that minimizes the rate of classification error. For our research purposes, false positives are more undesirable than false negatives. In order to have something meaningful to say about the frequency with which authors (or research teams) use recycled material, it is better to miss some cases which on inspection might be labeled as TR than to have the algorithm count as TR instances that we would not judge to be so. As a result, we need to ensure our strategy is especially accurate in making positive classifications.

Our score did not capture every feature of the Levenshtein distance matrix, in part because additional measures were more difficult to associate with observed practices such as copy-paste duplication and patchwriting. While we could have examined features like the ratio of matched to unmatched words in sentence pairs, our preliminary testing of the algorithm revealed that such scores can often be misleading in technical writing. Sentences in some academic settings can be dozens of words long, complicating efforts to use such ratios for analysis. Methods that inflate or discount the relative impact of different parts of speech can also be misleading in academic language. In our experience, relatively parsimonious methods seem to work best when dealing with technical writing.

4.7 Validation

To validate the method, we applied the algorithm to a sample of papers from our dataset.[11] We included five grants from a variety of disciplines, resulting in 10 pairwise comparisons per grant and 50 total comparisons. We produced scores, as described above, for each sentence pair in this set.[12] Next, we read through sentence pairs in the output and manually coded pairs as *recycled* or *not recycled* based on our intuitive sense of whether the amount and kind of overlap shown could

[11] A full description of the process used to determine the dataset is beyond the scope of the present article but will be described fully in a subsequent publication; the dataset included five articles from each of 10 grants across four STEM disciplines representing four of seven NSF directorates. Each discipline included two sub-disciplines (e.g., Engineering: Civil, Mechanical, and Manufacturing Innovation, or CMMI, and Electrical, Communications, and Cyber Systems, or ECCS).

[12] Human coding was performed by a single coder with no replication. Future efforts at validation will employ three coders, each coding overlapping subsamples to generate measures of intercoder reliability.

reasonably have been due to chance or use of commonplaces, or if it seemed likely that the overlap was the result of reusing specific material from the prior work.[13]

As one might imagine, the vast majority of sentence pairs have very low similarity scores—often close to zero. In preliminary tests, which included a sample of five papers by the same research team, we saw no pairs with scores below 7 that we judged to be instances of TR. Based on these early tests, we employed a cutoff TR score of 7.0 to facilitate the hand-coding labor—a strategy that reduced the number of sentence pairs needing hand-coding from tens of thousands to 303.

After performing the hand coding, we carefully analyzed the relationship between the machine TR score and our hand coding. In order to choose a cutoff value that we could use to classify other texts, we ran an iterative maximum likelihood function that minimized the error rate of the classifier. This function yielded a local minimum at a score of 9.599. We rounded this score to 9.6 and, as demonstrated in Figure 3, plotted the machine scores to determine how many false positives and negatives would result.

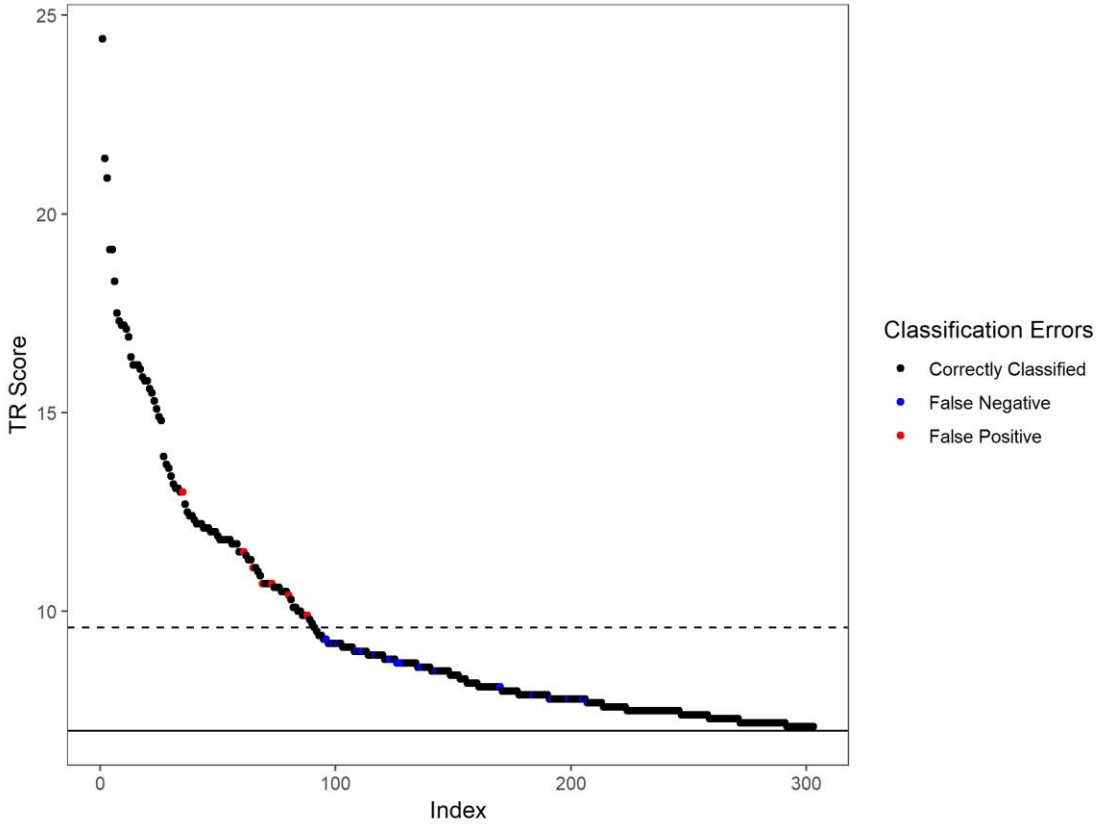

Figure 3. TR algorithm scoring vs. human scoring, training set (N = 303).

[13] We should note that our scoring does not take into account whether the new work included a citation or other attribution to the prior work. While citing one's prior work is often seen as a necessary condition for the ethical use of TR, our research aims at this point do not include assessing such matters.

5.0 Results

Figure 3 demonstrates that out of 303 hand-coded sentences, the classifier identified 7 false positives and 25 false negatives. The remaining 76 positive sentences and 193 negative sentences were correctly classified. The result was a precision score of 92.77 and a recall score of 74.04. The F1 score, which is a combined measure of precision and recall, was 82.35. While our weak recall score caused F1 to fall below the 0.90 threshold, we believe that a strong precision score (based on the accuracy of identifying true positives) shows reasonably effective performance. As we are interested in using the algorithm to make comparisons across a variety of document types rather than to definitively identify all instances of TR, this tradeoff is acceptable for our purposes. In the future, we hope to determine whether TR is more prevalent in some fields and subfields than others. In this sense, missing out on a few instances of genuine TR was far less important than frequently misidentifying TR.

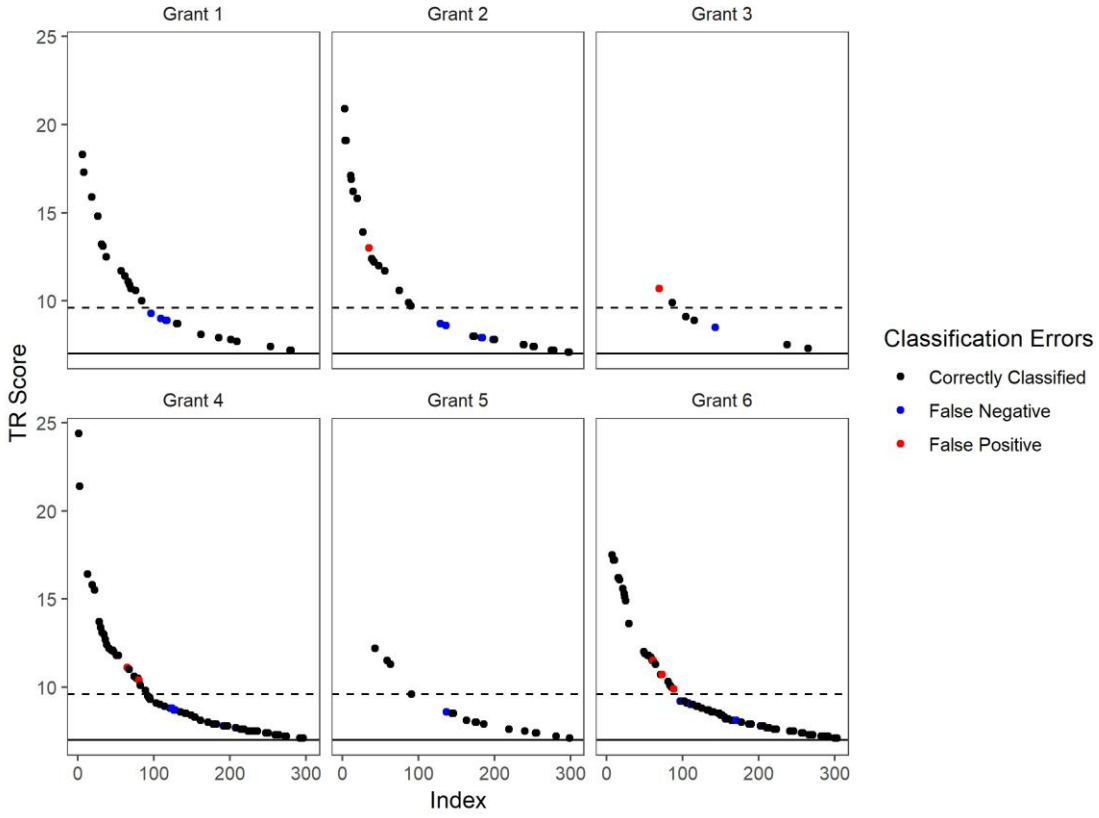

Figure 4. TR score vs. index by academic subject, training set (N = 303).

Given these aims, however, it was also important to evaluate whether training set performance was better or worse for different research fields. Our hand-coded sample included papers from a variety of academic disciplines, including biology, engineering, and sociology. Figure 4 shows classification errors by grant. We have de-identified the grants in order to

maintain the anonymity of the researchers. However, it is not necessary to identify which grant belonged to which field since the results demonstrate that classifier accuracy was not dependent upon grant.

This is a promising sign given that both the structure and prose used in research reports in these disciplines can differ substantially (see Klebanov et al., 2018). Sociology papers, for example, typically included extensive theory and literature sections, whereas the engineering papers in our dataset often included a great deal of information about methods, materials, and protocols.

To more robustly investigate the effects of article features on sentence classification accuracy, we use logistic regression techniques. We include two relevant features: the grant to which a sentence belonged and the position of the sentence within the paper in which it was written. This second term helps us to evaluate whether errors were occurring earlier or later in papers on average. Earlier errors might tell us that we are worse at predicting errors in introductions, literature reviews, or other early stage aspects of research papers. Later errors might tell us that we have failed to correctly identify TR in the case of results and discussions.

Figure 5 presents logistic regression coefficients. These coefficients tell us whether relevant text and document attributes have an impact on the accuracy of our classifier. If any coefficient is significant, it would suggest that the algorithm does a better job of classifying TR in some texts than others, or of identifying instances of TR in certain parts of scholarly work (such as the methods section, the results section, or conclusions). The results, however, are encouraging: we see no evidence of significant effects. All the horizontal bars in the figure, which represent 95% confidence intervals surrounding the coefficient estimates, contain zero. While this test is rudimentary, it helps assuage concerns that our method is only adept at classifying TR in specific fields of research or in certain sections of research reports.

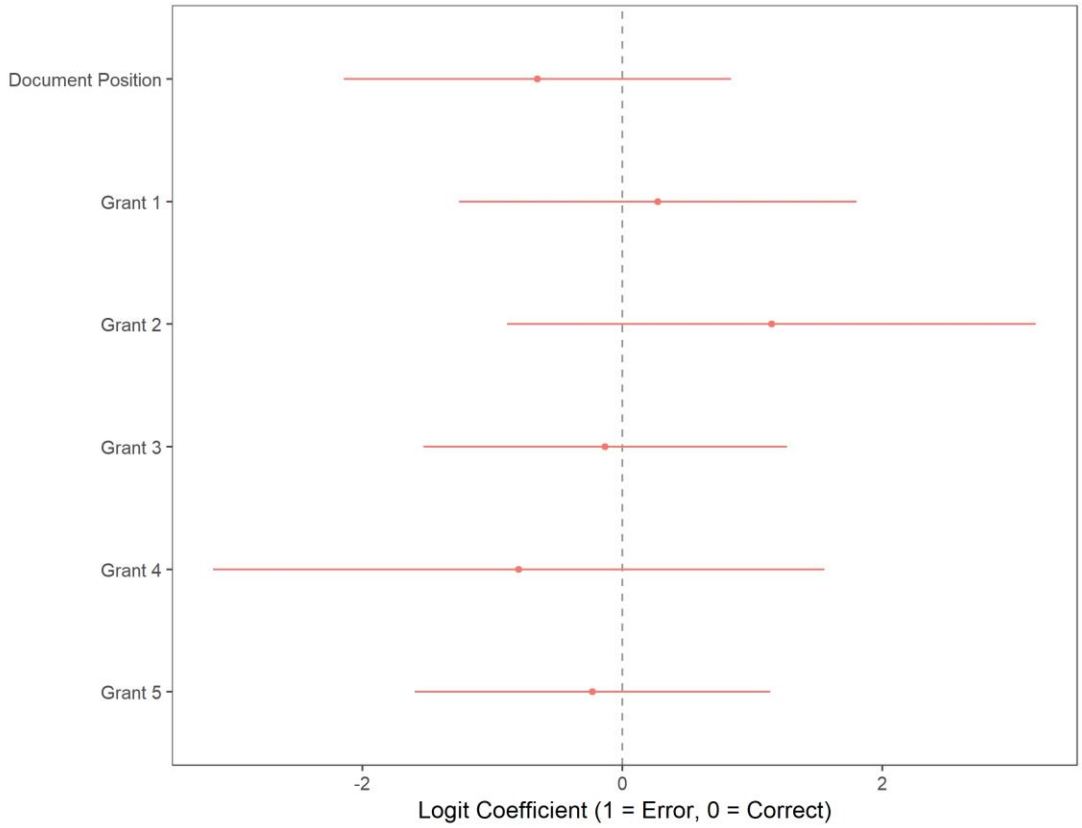

Figure 5. Beta coefficients, logistic regression model predicting coding errors (*N* = 303).

6.0 Discussion

Together, these results suggest that the algorithm does an acceptable and unbiased job of classifying true TR at the sentence level. As a result, we could be fairly confident that a count of sentences exceeding scores of 9.6 on the sentence classifier will be a meaningful indication of the extent of document-level TR. Using these scores, we will be able to aggregate our findings to the grant level, and ultimately investigate variables such as discipline and year. Because these sentence scores come from a sample of articles within a broader set of disciplinary publications, their occurrence is also probabilistic. Instances of TR as identified by our algorithm will likely follow a Poisson-like zero-inflated probability distribution with corresponding sample variance. These assumptions will facilitate statistical comparison when we engage in an analysis of the full sample of texts described above.

7.0 Conclusion

The system we have developed offers the possibility of analyzing large corpora of texts produced by the same author(s) in order to determine the types, nature, and extent of TR across those corpora. We stress the importance of taking a neutral stance toward the practice of TR that might be identified by this system: discourse communities (see Swales, 2017) define and determine the

norms of written and oral communication uniquely and as a function of their genres, cycles of credit and credibility, and rhetorical purposes. As stated earlier, our goal in creating a system of identifying TR is to facilitate the further analysis of this phenomenon in an attempt to extend our theories of written communication across a range of academic and professional settings.

8.0 Directions for Future Research

The decision to include a wide range of STEM fields had implications for developing our method of analysis. In particular, we needed to account for the fact that the standard structure of research reports in some social science disciplines (and even some STEM subfields) does not follow the archetypal IMRD (Introduction, Methods, Results, and Discussion) format common in much STEM research communication. As a result, our analysis does not attempt to map quantity or characteristics of TR on the IMRD structure. Given that much of the discourse and many of the guidelines for TR in STEM view recycling in some sections (especially methods sections, but occasionally introductions as well) as being more appropriate than recycling in other parts of such papers, it would be useful to see how recycling practices align with these structures.

In addition, it will be important to test the system on other kinds of corpora beyond research articles. The creation of separate, genre-sensitive algorithms may be required to study TR in a variety of contexts, rather than assuming that a single algorithm can be used universally across large corpora. While we have analyzed published articles from a variety of academic fields and subfields in this study, we have made several restrictive assumptions regarding our case selection. NSF-funded STEM proposals are likely to yield papers published in highly visible peer-reviewed journals. However, academic knowledge production and dissemination is becoming increasingly diverse, with the development of pre-print repositories, pre-analysis plans, open-access journals, and other modalities for rapid online content delivery. We know very little about TR in these contexts, much as we know very little about the format of content on these platforms in the first place. We hope to continue our investigation of TR in a way that accommodates these theoretical and practical considerations. So too do we hope to ascertain in the near future whether the methods we have introduced in this paper will be applicable to these diverse contexts.

Finally, further research could explore the relationship between TR of alphabetic text and other media included in research articles, such as graphs, charts, photographs, and illustrations. Although the system we have described here is not capable of identifying the replication of such material, it may be possible to study the relationship between groups of articles with high levels of TR as a subset and subject them to human scrutiny for the replication of visuals. Such analyses could extend our understanding of writers' constructs of the media of communication and the norms and practices that govern how such media are used across their publications.

Author Biographies

Ian G. Anson is Assistant Professor of Political Science at the University of Maryland, Baltimore County (UMBC). His recent publications have spanned topics in public opinion,

political behavior, the scholarship of teaching and learning, and quantitative research methods. Ian holds a Ph.D. in political science and an M.S. in applied statistics from Indiana University. A complete biography is available at www.iananson.com.

Cary Moskovitz is Professor of the Practice and Director of Writing in the Disciplines in the Thompson Writing Program at Duke university. Cary holds a Ph.D. in aerospace engineering from North Carolina State University and Master of Architecture from Virginia Tech. His articles and essays related to writing pedagogy and text recycling have appeared in such publications as *The Chronicle of Higher Education*, *Science*, *College Composition and Communication*, *Research Integrity and Peer Review*, and *Advances in Engineering Education*. He has served as a consultant on writing pedagogy and led faculty workshops at a number of U.S. colleges and universities. Cary is Principle Investigator for the Text Recycling Research Project and directs the Duke Reader Project.

Chris M. Anson is Distinguished University Professor and Director of the Campus Writing and Speaking Program at North Carolina State University. He has published 17 books and over 140 articles focusing on written communication. Before joining NC State in 1999, he was Morse Alumni Distinguished Professor at the University of Minnesota, where he also directed the Program in Composition and Communication for 9 of his 15 years there. His online c.v. is located at www.ansonica.net.

References

Alzahrani, S. M., Salim, N., & Abraham, A. (2011). Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(2), 133–149.

Andreescu, L. (2013). Self-plagiarism in academic publishing: The anatomy of a misnomer. *Science and Engineering Ethics*, *19*(3), 775–797.

Beversluis, E. (2016, 30 Nov.). "Self-Plagiarism"? You gotta be kidding. Retrieved from https://writingcooperative.com/self-plagiarism-you-gotta-be-kidding-746c3c58921c

Blei, D. M, Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research 3*(1), 993–1022.

Bouville, M. (2008). Plagiarism: Words and ideas. *Science and Engineering Ethics*, *14*(3), 311–322.

Bretag, T., & Carapiet, S. (2007). A preliminary study to identify the extent of self-plagiarism in Australian academic research. *Plagiary: Cross-Disciplinary Studies in Plagiarism Fabrication, and Falsification*, *2*(5), 1–12.

Bretag, T., & Mahmud, S. (2009). Self-plagiarism or appropriate textual re-use? *Journal of Academic Ethics*, *7*(3), 193–205.

Butakov, S., & Scherbinin, V. (2009). The toolbox for local and global plagiarism detection. *Computers & Education*, *52*(4), 781–788.

Collberg, C., & Kobourov, S. (2005). Self-plagiarism in computer science. *Communications of the ACM*, *48*(4), 88–94.

Collberg, C., Kobourov, S., Louie, J., & Slattery, T. (2003). SPlaT: A system for self-plagiarism detection. *Proceedings of the IADIS International Conference on WWW/Internet*, 508–514.

Evola, M. (2018, 10 Oct.) Self-plagiarism: Can you steal from yourself? Retrieved from https://www.depts.ttu.edu/research/scholarly-messenger/2016/October/rcr-self-plagiarism.php

Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, *21*(3), 267–297.

Gupta, D. (2016). Study on extrinsic text plagiarism detection techniques and tools. *Journal of Engineering Science & Technology Review*, *9*(5).

Hiremath, S. A., & Otari, M. S. (2014). Plagiarism detection-different methods and their analysis. *International Journal of Innovative Research in Advanced Engineering*, *1*(7).

Horbach, S. P. J., & Halffman, W. (2019). The extent and causes of academic text recycling or 'self-plagiarism.' *Research Policy*, *48*(2), 492–502.

Howard, R. M. (1999). *Standing in the shadow of giants: Plagiarists, authors, collaborators.* Stamford, CT: Ablex.

Hwang, E. S. (2017). On difficulty in handling text recycling. *Science Editing*, *4*(2), 86–88.

Israel, M. (2019, Jan. 19) Self-plagiarism? When re-purposing text may be ethically justifiable. Retrieved from https://ahrecs.com/research-integrity/self-plagiarism-when-re-purposing-text-may-be-ethically-justifiable

Klebanov, B. B., Priniski, S., Burstein, J., Gyawali, B., Harackiewicz, J., & Thoman, D. (2018). Utility-value score: A case study in system generalization for writing analytics. *Journal of Writing Analytics*, *2*, 314–328. Retrieved from https://wac.colostate.edu/docs/jwa/vol2/klebanov.pdf

Li, Y., & Bo, L. (2007). A normalized Levenshtein Distance Metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6),1091–5.

Meuschke, N., & Gipp, B. (2013). State-of-the-art in detecting academic plagiarism. *International Journal for Educational Integrity*, *9*(1).

Moskovitz, C. (2018). Text recycling in scientific writing. *Science and Engineering Ethics*. DOI: 10.1007/s11948-017-0008-y

Mudrak, B. (2017). Self-plagiarism: How to define it and why you should avoid it. Retrieved from https://www.aje.com/arc/self-plagiarism-how-to-define-it-and-why-to-avoid-it/

Murray, J. (2009). *Non-discursive rhetoric: Image and affect in multimodal composition*. Albany: State University of New York Press.

Rish, I. (2001). An empirical study of the naive Bayes Classifier. *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, *3*(41–46), 22.

Roig, M. (2005). Re-using text from one's own previously published papers: An exploratory study of potential self-plagiarism. *Psychological Reports*, *97*(1), 43—49. https://doi.org/10.2466/pr0.97.1.43-49

Roig, M. (2008). The debate on self-plagiarism: Inquisitional science or high standards of scholarship? *Journal of Cognitive and Behavioral Psychotherapies*, *8*(2), 245–258.

Roig M. (2015). *Avoiding plagiarism, self-plagiarism, and other questionable writing practices: A guide to ethical writing.* ORI - The Office of Research Integrity. Retrieved from https://ori.hhs.gov/plagiarism-16a

Scanlon, P. M. (2007). Song from myself: An anatomy of self-plagiarism. *Plagiary: Cross-Disciplinary Studies in Plagiarism, Fabrication, and Falsification*, 57–66.

Schein M., & Paladugu, R. (2001). Redundant surgical publications: Tip of the iceberg? *Surgery*,*129*(6), 655–661.

Schneider, L. (2016, 11 Apr.). Academic self-plagiarism: Misconduct or a literary art form? *For Better Science*. Retrieved from https://forbetterscience.com/2016/04/11/academic-self-plagiarism-misconduct-or-a-literary-art-form/

Schryer, C. F. (2011). Investigating texts in their social contexts: The promise and peril of rhetorical genre studies. In D. Starke-Meyerring, A. Paré, N. Artemeva, M. Horne, & L. Yousoubova (Eds.), *Writing in knowledge societies* (pp. 31–52). Anderson, SC: Parlor Press.

Su, Z., Ahn, B., Eom, K., Kang, M., Kim, J., & Kim, M. (2008). Plagiarism detection using the Levenshtein Distance and Smith-Waterman Algorithm. *2008 3rd International Conference on Innovative Computing Information and Control*, 569–69. IEEE.

Sun, Y. C. (2013). Do journal authors plagiarize? Using plagiarism detection software to uncover matching text across disciplines. *J. Eng. Acad. Purposes, 12*(4), 264–272. DOI: 10.1016/j.jeap.2013.07.002

Swales, J. M. (2017). The concept of discourse community: Some recent personal history. *Composition Forum, 37*.Retrieved from https://compositionforum.com/issue/37/swales-retrospective.php

Tao, G., Guowei, D., Hu, Q., & Baojiang, C. (2013, Sept.). Improved plagiarism detection algorithm based on abstract syntax tree. *Fourth International Conference on Emerging Intelligent Data and Web Technologies* (pp. 714–719). IEEE.

Thurman, R. H., Chervenak, F. A., McCullough, L. B., Halwani, S., & Farine, D. (2016). Self-plagiarism: A misnomer. *American Journal of Obstetrics & Gynecology*, *214*(1), 91–93.

Trim, C. M., & Flammer, S. M. A. (2017). Managing text-recycling: An ongoing issue. *Veterinary Anaesthesia and Analgesia*, *44*(4), 695–696.

Turnitin. (2016, July 20). Is recycling your own work plagiarism? Retrieved from https://www.turnitin.com/blog/is-recycling-your-own-work-plagiarism

Walczak, S. (2017). A text analytic approach to classifying document types. *Journal of Writing Analytics, 1*, 103–146. Retrieved from https://wac.colostate.edu/docs/jwa/vol1/walczak.pdf

Warner, A. B. (2007). Constructing a tool for assessing scholarly webtexts. *Kairos*, *12*(1).

White, D. R., & Joy, M. S. (2004). Sentence-based natural language plagiarism detection. *Journal on Educational Resources in Computing (JERIC)*, *4*(4), 2.

Young, L., & Soroka, S. (2012). Lexicoder sentiment dictionary. Retrieved from www.lexicoder.com

Yousuf, S., Ahmad, M., & Nasrullah, S. (2013, October). A review of plagiarism detection based on Lexical and Semantic Approach. In *2013 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA)* (pp. 1–5). IEEE.

Zini, M., Fabbri, M., Moneglia, M., & Panunzi, A. (2006). Plagiarism detection through multilevel text comparison. *2006 Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (Axmedis'06)*, 181–85. IEEE.