International Journal of Communication 13(2019), 3935–3954 1932–8036/20190005

Toward Open Computational Communication Science: A Practical Road Map for Reusable Data and Code

WOUTER VAN ATTEVELDT[1]
Vrije Universiteit Amsterdam, the Netherlands

JOANNA STRYCHARZ
DAMIAN TRILLING
University of Amsterdam, the Netherlands

KASPER WELBERS
Vrije Universiteit Amsterdam, the Netherlands

Computational communication science (CCS) offers an opportunity to accelerate the scope and pace of discovery in communication research. This article argues that CCS will profit from adopting open science practices by fostering the reusability of data and code. We discuss the goals and challenges related to creating reusable data and code and offer practical guidance to individual researchers to achieve this. More specifically, we argue for integration of the research process into reusable workflows and recognition of tools and data as academic work. The challenges and road map are also critically discussed in terms of the additional burden they place on individual scholars, which culminates in a call to action for the field to support and incentivize the reusability of tools and data.

Keywords: computational communication science, open science, reproducibility, reusability, workflow

Many scholars in communication science are turning to computational methods to gather and analyze large data sets of digital traces of human communication (e.g., Cappella, 2017; Shah, Cappella, & Neuman, 2015; Trilling, 2017; van Atteveldt & Peng, 2018). Computational communication science (CCS) has the potential to greatly accelerate the pace of discovery in the field. As argued by González-Bailón

Wouter van Atteveldt: wouter@vanatteveldt.com
Joanna Strycharz: J.Strycharz@uva.nl
Damian Trilling: d.c.trilling@uva.nl
Kasper Welbers: k.welbers@vu.nl
Date submitted: 2018–10–03

[1] We wish to thank the editors and reviewers for their many comments and suggestions. We would especially like to thank Jef Ausloos of the University of Amsterdam for advice and information regarding legal aspects.

(2017), the availability of new types of data and computational methods makes it possible to explore and empirically test ideas that were unable to be tested with classical methods. The move from relatively small studies relying on self-reports in artificial settings to analyzing large amounts of digital traces of actual behavior in its social context allows us to model communication and social behavior in unprecedented detail (van Atteveldt & Peng, 2018). At the same time, valid criticism can be made regarding the reliability, validity, and reproducibility of CCS (e.g., boyd & Crawford, 2012; Chang, Gerrish, Wang, Boyd-Graber, & Blei, 2009; Grimmer & Stewart, 2013; Hargittai, 2015; Kitchin, 2014; Lazer et al., 2009; Stockemer, Koehler, & Lenz, 2018; Wallach, 2016).

Of course, questions of reproducibility and validity are not limited to computational communication research, as shown by recent calls for transparent and reproducible research in what is called the open science movement (e.g., Bowman & Keene, 2018; Klein et al., 2018; Munafò et al., 2017; Nosek et al., 2015). In general, the key to fostering open science is data access, transparent design, and analytical transparency—that is, the sharing and citing of research data, code, materials, and designs (Munafò et al., 2017; Nosek et al., 2015). As argued by Klein and colleagues (2018), the availability of data and analysis scripts fosters the analytic reproducibility of results, or the ability to reproduce the exact analyses reported in a paper. This, in turn, allows us to judge the analytic robustness of findings, or the extent to which results are based on particular choices in processing analysis, as well as their replicability and generalizability, or the extent to which similar results can be achieved by replicating the study with new data gathered in a similar or different context, respectively.

The main argument of this article is that CCS is in a key position to foster open communication science by creating a culture and practice of reusable data and code. Just as CCS itself is made possible by the confluence of the availability of digital communication traces, analysis tools, and computing resources (van Atteveldt & Peng, 2018), these factors also make it possible to conduct computational research in a reproducible and reusable way. Having "born-digital" data makes it easier and more meaningful to share the data; most analyses are driven by open-source scripts or scriptable tools; and freely available computing resources make it much more convenient to share data and code.

We understand reusability in a broad sense that implies transparency and reproducibility. Sharing the code used for data gathering, cleaning, and analysis makes research fully transparent and, given that data are openly accessible, directly reproducible. In addition, reusability means that data and code are structured, stored, and documented in such a way as to allow and encourage other scholars to adapt them to their specific needs. In this sense, reusability goes beyond strict reproducibility or replicability.

Recent years have seen the growth of an impressive body of literature discussing the possibilities and challenges in computational social science in general (e.g., Alvarez, 2016; Lazer et al., 2009; Salganik, 2017) and in computational communication science specifically (e.g., Cappella, 2017; van Atteveldt & Peng, 2018). This article contributes to that literature by focusing on the reusability and reproducibility of CCS research rather than on specific substantive or methodological opportunities or challenges. By viewing computational research through the lens of the open science movement, this article describes how CCS can foster openness in communication research. Thus, although our main argument directly speaks to the accessibility, reliability, and validity challenges identified by van Atteveldt and Peng (2018, pp. 85–87), we

focus on how reusability can help tackle these challenges rather than suggesting immediate methodological improvements. Similarly, rather than directly contributing to the substantive research questions implied in the four "vectors into the future" posited by Cappella (2017), we argue that the reusability of code and data is needed to attain the scale and methodological sophistication needed to further research in these vectors.

This article further contributes to the emergence of open (computational) communication science by drawing a practical road map for reusable data and code for computational researchers. This map lists concrete suggestions and tools for reusability practices in each step of the research process. These practices are relatively easy to implement and can directly benefit individual researchers and labs by increasing efficiency and visibility (cf. Klein et al., 2018).

Before discussing the road map, we will present the specific goals and challenges related to the reusability of data and code. Finally, in the concluding section, we discuss how individual researchers can overcome the remaining challenges they face and what can be done to foster a culture of reusability (and actual reuse) of data and tools.

Goals and Challenges for the Reusability of Data

Data play a central role in most (empirical) research, but they are especially important in computational communication science. Data sets in CCS are often relatively large and heterogeneous and require substantial effort and technical skills to gather, clean, and integrate. Sharing these data sets can improve the scientific return on this investment and increase the reproducibility of studies. Thus, one of the most important steps toward open computational science is to publish data sets so research can be reproduced and replicated and materials can be reused (Miguel et al., 2014).

Such data should be published under the FAIR (findable, accessible, interoperable, and reusable) guidelines (Wilkinson et al., 2016). Findable and accessible data are properly licensed and published in a trusted repository that can be searched and indexed. Interoperability and reusability require that data are stored in an open (nonproprietary) format and include metadata on what the data mean and how they were collected (Klein et al., 2018). In moving toward reusable data, however, scholars need to deal with the ethical and legal challenges related to the use and sharing of proprietary and privacy-sensitive data.

Proprietary data, such as newspaper articles or collections of data in digital archives, are often protected by copyright, database, and contract law. Copyright serves to protect owners of the original content, such as news organizations or journalists. Database law extends copyright by protecting the creators of collections of content, even if the content itself is not copyrightable (e.g., naked facts). Contract law applies if the collection and use of data involve contractual agreements, such as licensing agreements or terms of service. For example, services such as LexisNexis and social media APIs require users to agree to terms of service before granting access, which can disallow publishing and even archiving the retrieved data. Thus, whether it is possible to use and share these data can depend on a complex interplay of laws and on the willingness of data owners to cooperate. Unfortunately, given the important role of (proprietary) data in the business models of the firms that own such data, including large technology firms and publishers,

it is unlikely that these companies will improve the access to and shareability of their data (Ekbia et al., 2015; Freelon, 2018).

It is worth noting that there are exceptions to copyright that enable the use of material for scientific research. The United States follows the fair use doctrine, which states that the use of copyrighted material for certain purposes, including criticism and research, does not constitute copyright infringement. Similarly, the European Union Copyright Directive (2001/29/EC) allows exceptions for scientific research, though member countries can autonomously decide which exceptions to acknowledge. The proposed EU directive on Copyright in the Digital Single Market might include specific exceptions for text and data mining for scientific research. However, these provisions are limited and often uncertain in scope, and open data publication of copyrighted material can infringe copyright even if the goal is to advance scientific research.

In addition to data ownership, there can be legal and/or ethical restrictions for publishing and sharing data sets that contain personal data, defined broadly as any information that relates to an identified or identifiable individual. Besides data from surveys and human-subject experiments, this type of data also applies to many digital traces such as social media posts and online behavior, which have become prime objects of inquiry for CCS. Such data can be highly sensitive; even seemingly innocuous data such as Facebook likes can be used to predict a large range of private information, including political views and sexual orientation (Kosinski, Stillwell, & Graepel, 2013).

Finally, as personal data have become more abundant—and, in many cases, more intrusive—we are witnessing a shift toward stronger regulation of the use of personal data. In the European Union, the General Data Protection Regulation (GDPR) puts strict limitations on storing and processing personal data, especially with sensitive data such as ethnicity and political opinions. Other jurisdictions, including several U.S. states, are also considering similar legislation to safeguard data rights. Although the GDPR specifically recognizes the importance of personal data in scientific research, it remains vague on how data rights should be achieved in practice, and national laws have not helped much in this regard.

A secondary problem related to both challenges is that laws and regulations relating to copyright and privacy can change and are often complex, they might differ among jurisdictions, and new regulations can be unclear and untested. For example, the scope of the GDPR is limited to research carried out in the European Union or on EU subjects, but it can also put limitations on storing and sharing data outside the European Union and on non-EU researchers working with EU subjects. As yet, it is unclear what these new and upcoming laws and regulations mean for academic research, especially for collaborations between EU members and non-EU members.

Goals and Challenges for the Reusability of Code

Next to the sharing and reusing of data, the reusability of code is central to open CCS. Rather than developing new code from scratch, researchers should, where possible, build on established (open-source) software. In addition to preserving their own time and resources, this contributes to the development of better tools, with tried and tested features and documentation, and collaboration in detecting and solving

bugs. Thus, the reusability of tools is a central concern in advancing and developing a more open computational communication science.

For reproducibility and replicability of research, it is vital that the materials and analytic procedures are transparent (Miguel et al., 2014; Nosek et al., 2015). For others to evaluate the correctness, robustness, and generalizability of their analysis, scholars should "offer a full account of the procedures used to collect or generate data" (referred to as production transparency) and "of how they draw their analytic conclusions from the data" (analytic transparency; American Political Science Association, 2012, p. 10). By sharing the actual code used to gather, clean, and analyze the data, other researchers can simply run the code again on the shared or publicly available data and arrive at the same results. This code should be posted to a public repository, and ideally a reviewer should check whether the code indeed allows the published results to be reproduced (Nosek et al., 2015, p. 1424).

While sharing the exact code used in analysis guarantees reproducibility, additional steps are required to improve reusability. Other scholars need to be able to adapt the code for use in different contexts and on different data sets, so the code must be properly cleaned and documented. Furthermore, if a new type of method is used, it can be valuable if the main functionality of the code is made more generally applicable instead of only working for the researcher's specific use case. This enhanced code can then be packaged and distributed as an open-source tool.

The main challenge for fostering reusable code is that these additional steps can be very time-consuming, and, unlike the development of the original code used in the analysis, this time does not directly contribute to the developer's own research. Thus, it is important that tools are identified and credited as academic work. However, software development, and especially maintenance and support, is often not seen as a stand-alone academic contribution. This can make investment in software a liability in the pursuit of an academic career, where impact measures are critical in the competition for tenure and grants.

One solution that is often mentioned for incentivizing code (and data) sharing is publishing a specialized research article to describe the code (or data set) that can then be used to cite it as an academic publication (Chavan & Penev, 2011). However, while this is a good idea for data or tools that form a clear contribution to the field, it merely shoehorns the contribution into the existing publication system. Many journals do not have a suitable format for discussing issues and choices related to data gathering or software design. Moreover, it is difficult to publish incremental new versions of software, and especially contributions to existing software. Ironically, this can make it less attractive to reuse and credit other people's software, because it can be easier to get credit for writing a new tool than for contributing to existing software. Over the long run, however, the field will benefit more from a smaller number of programs and data sets that are maintained by multiple scholars than from a plethora of one-off and often insufficiently maintained tools and data sets.

A Practical Road Map for Reusable Data and Tools in CCS

Although some of the challenges outlined above will require concerted and long-term action throughout the field, individual researchers and research groups can achieve an immediate improvement by

adopting a set of comparatively easy practices. This section argues that researchers can and should start developing and using reusable workflows.

A workflow is the sequence of steps taken to complete a process. Documenting and sharing a workflow accelerates and improves scientific progress (Gil, Groth, Ratnakar, & Fritz, 2009; Ludäscher et al., 2006). A major boon of computational research is that workflows can be made very explicit by sharing the code and software. Given access to the same data, most, if not all, steps of the analysis can be reproduced. Furthermore, if the code is sufficiently documented and modularized, others can not only reproduce the results from the original publication but also reuse the workflow in new projects (Hutton et al., 2016). Although some disciplines use dedicated scientific workflow systems (e.g., Kepler; https://kepler-project.org/), for CCS the more immediate concern is to establish good practices for developing workflows. To this end, we first discuss guidelines for developing CCS workflows divided into the four components shown in Figure 1: (1) research design, (2) data gathering, (3) data cleaning and analyses, and (4) dissemination.

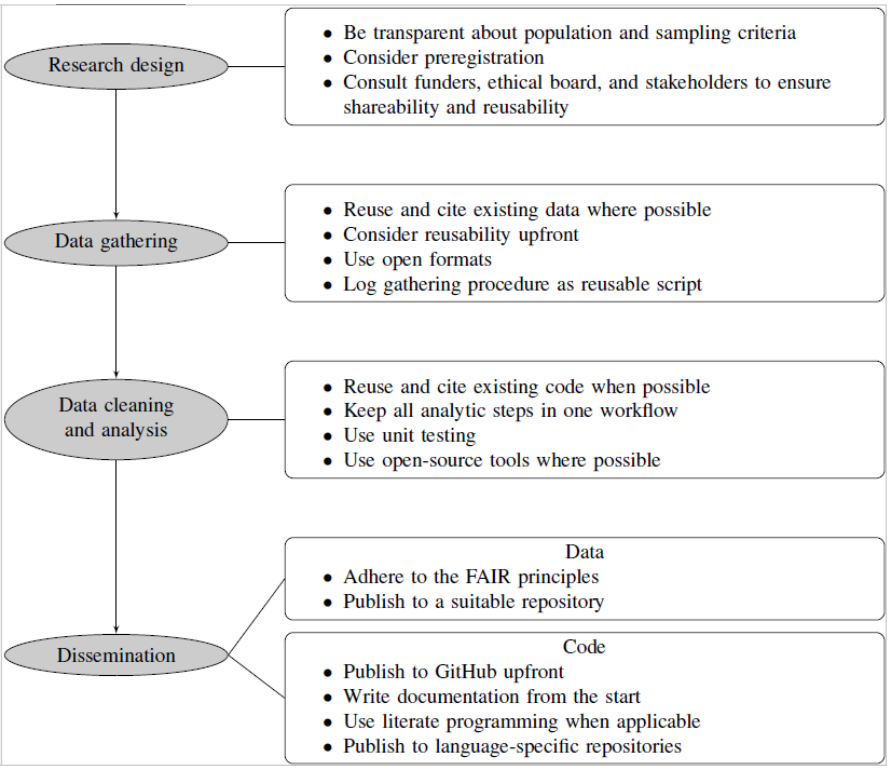

Figure 1. Considerations for an open Computational Science workflow.

Although we present the necessary considerations following the traditional social-scientific research process, it is important to note that many of these steps are easiest and most effective if adopted from the outset and in a systematic manner. For example, one can use GitHub, a popular repository hosting service, to publish code. In such a case, it is beneficial to place all code there from the start; this way, GitHub can be used to update the code and keep track of all changes to it as well as to distribute the code. Similarly, it is easier to structure and document data during the collection process than to bring it into shape later.

Research Design

Open CCS begins at the research design stage. Transparency about population and sampling criteria, while being important criteria for any social-scientific study, gains extra relevance for CCS. If these criteria are not well documented, code that is produced later may become essentially meaningless. Second, to increase transparency, we advise preregistration of studies when applicable. Preregistration involves a specification of hypotheses, methods, and analyses that are formally registered on a public website. They receive a time stamp and can be viewed by the scientific community (see Klein et al., 2018). Most important at the research design stage is to consult funders, ethical boards, and stakeholders to ensure the shareability of code and data that are produced later. Communicating the intention to share early in the process ensures that the right confidentiality and usage agreements can be made and that sharing is performed diligently.

Data Gathering

Before gathering data, researchers should check whether existing data can be reused (and cited) instead. If researchers decide to collect their own data, they need to consider the reusability and shareability of the resulting data. For instance, if the data-gathering process involves the collection of sensitive data that would prevent sharing, researchers should make sure that the data are anonymized from the beginning. A key for deanonymization should be kept in a separate place and should not normally be accessible even to the researchers.

Researchers also need to ensure that the provenance of the data is clear. It is essential to precisely log and document how the data were gathered and processed. For computational research, this generally means that the data should be accompanied by the script that was used to gather the data before substantive analysis. Ideally, these data contain the API calls that were used to gather data (if applicable), even if the data cannot be replicated purely with those calls (because of access rights or a dynamic data source). This step not only increases the transparency of the data-gathering process but also allows researchers to update the data set with new data. In short, avoid any manual steps in the data-gathering process.

We also recommend using open formats such as CSV, JSON, or XML (see also Healy, 2018) rather than proprietary formats such as Microsoft Excel or IBM SPSS. Open formats ensure that data can be accessed in the future and without the need to buy proprietary software (Günther, Trilling, & Van de Velde, 2018). Where possible, data in proprietary formats should be converted directly so all cleaning and analysis operations can use the open format. Unless specifically needed, it is also advisable to use CSV or JSON rather than open software-specific formats such as RDS (R) or pickle (Python) for all nontemporary data files.

Data Cleaning and Analysis

For transparency and replicability, it is important that the exact code that is used for analysis is published. It is highly advisable that all steps are implemented in a single computing environment. Ideally, the code can be run in one step without manual intervention or transferring data between applications. In particular, any cleaning or recoding of the raw data should be done programmatically rather than by editing the data file manually. For example, a separate CSV file that contains the needed fixes can be made, and then an R or Python script can be used to apply these fixes. This process ensures that it is obvious what changes were made to the data after collection.

A complete guide to writing reusable code is beyond the scope of this article. Fortunately, many practical guides can assist scholars in ensuring that their cleaning and analysis code is reusable. For instance, Healy (2018) provides an introduction to tools that can be used to create workflows that can be rerun with a single command, and Gandrud (2015) explains in detail how to use R with version control—in particular, to manage different versions of files using a platform such as GitHub.

Data Dissemination

The use of large, heterogeneous, and preexisting data is a defining aspect of computational communication research, and the emergence of large digital data sets was a central component in the emergence of computational methods (van Atteveldt & Peng, 2018). Many journals now require the data used in a study to be published on a repository such as DataVerse together with the article, but at least two more steps must be taken to achieve reusability rather than mere reproducibility.

First, it is important to structure the data in a way to maximize reusability. Although multiple standards and checklists exist, a commonly used standard is that of FAIR data—or findable, accessible, interoperable, and reusable (FORCE11, 2016). Practically, this means that data should be published in a repository such as DataVerse, the data should be stored in an open format (such as CSV, JSON, or XML) and be accompanied by sufficiently rich metadata (and documentation) to interpret the data, and the metadata need to be accessible even if the raw data cannot be. By having data in a shareable format when one begins the research process ensures other researchers that the published code can be used on the data. Finally, to be reusable, the (meta)data should contain all relevant variables, not just the ones used in the related study, and the data should have a clear (preferably open-access) data usage license.

Before sharing, researchers need to consider privacy, copyright, and other legal issues that may hinder sharing of the data. These limitations can be mitigated in several ways. First, researchers should publicly share all data except for sensitive data. Any existing or generated metadata, annotations, or analyses of the data can generally be shared even if the raw data cannot. Such additional data should be linked to the original data with a unique identifier—either a generated identifier or an external ID such as a URL or, for example, a Twitter Status ID. The sensitive data can then be stored in a secure and trusted location, such as a university archive, where it is stored in an encrypted format and access to the file is restricted based on the sensitivity of the data (Bar-Sinai, Sweeney, & Crosas, 2016). By retaining a link to

this trusted location in the published data, it remains possible to consult the raw data of the study should there be a genuine need.

Another possible solution to privacy and copyright issues can be the use of nonconsumptive research practices (Zeng, Ruan, Crowell, Prakash, & Plale, 2014). This means that access to the data is given to other researchers without physically copying the data. In general, this approach can take two forms: First, researchers can be given access to the data sets through a website and/or an API that allows them to analyze the data with predefined algorithms without being able to view or download the sensitive data. For example, AmCAT (van Atteveldt, 2008) can grant nontrusted users access to article metadata and the query interface, allowing them to perform keyword queries and view aggregate results and snippets without giving access to the underlying data. The second option is to allow users to run their own analysis algorithms on the data through restricted physical or virtual access, but prevent them from copying the underlying data. For example, the Hathi Trust allows researchers to use a virtual machine with a data capsule. Briefly put, a researcher can design and test an analysis on a desensitized subset of the data as normal. Then the virtual machine switches to secure mode to run the analysis script on sensitive data. In secure mode, the virtual machine does not have general network access, and the (generally nonsensitive) analysis results are written to a file and checked before sending them back to the researcher. Although both solutions limit the flexibility (and convenience) of the user, they can allow for data to be made accessible in the face of privacy or copyright limitations.

Code Dissemination

At the risk of oversimplifying matters, we distinguish two types of products that can be disseminated: code and code snippets, which are mainly used to reproduce a given analysis, and full-fledged tools or applications.

Sharing the actual code of a study should not result in much extra work if the steps suggested above are followed (e.g., using GitHub or another version control system from the start). Disseminating this type of code to others mainly requires revising the documentation and ensuring that it is understandable for researchers outside the specific project. The next section describes a recommended setup for a research compendium to standardize such sharing.

Sharing reusable (general purpose) tools or applications, on the other hand, does require extra work but also lowers the burden for others to reuse them. If one documents and tests code properly while developing it, the extra effort is manageable. It is especially advised to use unit testing for reusable code. A unit test is essentially a small script that tests whether the code produces the expected result for given inputs. Both Python and R offer facilities to automatically run such unit tests, so if a future change breaks the code such that it no longer produces the expected result, then the user can be warned immediately instead of finding out much later. Publishing reusable code in an R package or Python module on the respective repositories (CRAN and PyPI) ensures that it integrates seamlessly in workflows others are using.

Research Compendiums for Disseminating Code and Data

We recommend creating a research compendium as a repository on a service such as GitHub for each research paper to disseminate both the code and the related data. As defined by Marwick, Boettiger, and Mullen (2018), such a compendium should "maintain a clear separation of data, method, and output, while unambiguously expressing the relationship between those three" and "specify the computational environment that was used for the original analysis" (p. 81).

The data part of such a compendium should distinguish among the following data types, for example, by creating separate subfolders: raw data obtained directly from an external source that can be published; private data that cannot be shared publicly (these can either be stored only locally while included in the repository in the .gitignore file or stored in the repository in encrypted form); intermediate data that is used in analyses and can be shared publicly, possibly including aggregated or anonymized versions of the private data; and final data that are meant for publication and reuse.

The code published in the compendium should contain at least the following: the data processing code that produces the intermediate data based on the raw and private data or public sources (e.g., APIs or data from DataVerse); and the analysis code, possibly in the form of notebooks that use the (intermediate) data to produce all reported results.

The results included in the compendium should contain any data sets that are considered final, which should include the appropriate metadata and documentation, and the results of the analyses—for example, the output of the analysis code. Additionally, the compendium should contain documentation in the form of, for example, a README file that describes the results, links to the related publication, and contains installation and build instructions; and license information for code, data, and results—preferably MIT or a similar permissive license for code and CC-BY for data and results.

Finally, to make a compendium easier for others to work with, it can contain unit tests that test whether the analyses can indeed be produced, at least without errors; a continuous integration file such as Travis (or equivalent) to automatically check the unit tests on each commit; a makefile or equivalent that automatically runs all data processing and analyses code; and a dockerfile or equivalent that automatically builds and packages the code and dependencies as a container. A container includes everything needed to run an application (i.e., the actual code as well as required libraries and system tools). This guarantees that a given container, run on different machines, will always produce the same results.

Such a compendium can serve as the online appendix to a paper, but by hosting it on a service such as GitHub, a researcher can update the code later while still preserving the original appendix as a named release (for reproducibility). While working on the paper, the repository can be kept private and then set to public when the paper is accepted. If desired (or required by the journal), raw data can also be moved to DataVerse on publication, with an added command at the start of data processing to download the data to the intermediate folder. By making explicit all the steps from raw data to analyses and making use of literate programming and services such as GitHub, Travis, and Docker, it becomes much easier for third

parties (including the "later you") to understand, scrutinize, and reuse (parts of) the code and data (Piccolo & Frampton, 2016).

There are existing implementations of such compendiums—for example, the rrtools package for R (Marwick, 2019). Our compendium setup also can be downloaded along the lines described earlier from https://github.com/ccs-amsterdam/compendium. Note that a compendium utilizes a number of external commercial services (such as GitHub, Travis, and Docker). However, the reusability does not depend on the specific service providers, and for each service, commercial and noncommercial alternatives exist (such as BitBucket, Circle-CI, and Singularity, respectively). The distributed nature of Git also means that it is trivial to keep local copies of the whole repository (including the version history), and a copy can easily be placed on a public resource such as Open Science Framework (https://osf.io).

Which Software to Use?

When implementing an open CCS workflow, one needs to decide which software packages to use. This section offers some practical advice on what packages to use. Ideally, workflows are integrated into a single software environment.

Programming languages such as R or Python lend themselves to this task, because they have a rich ecosystem of packages for a wide range of functionalities, from scraping and data cleaning to text analysis, machine learning, and statistical modeling.

Table 1 shows some widely used packages that fit an integrated workflow.

Table 1. An Overview of Popular Modules Fit for Developing Integrated Workflows in R or Python.

	R	Python
Data gathering		
Cleaning and preprocessing	tidyverse, data.table	pandas, numpy, scipy
Scraping	httr, rvest	requests, lxml
Data analysis		
Text analysis	quanteda, spacyr	nltk, spacy
Network analysis	igraph	igraph
Modeling	lme4, vars	statsmodels
Machine learning	CORElearn	scikit-learn
Dissemination		
Literate programming	RMarkdown	Jupyter Notebook

Note. For the sake of readability, we do not include citations in the table but provide them here: tidyverse (Wickham & Grolemund, 2017); data.table (Dowle & Srinivasan, 2017); httr (Wickham, 2018) ; rvest (Wickham, 2015); quanteda (Benoit et al., 2017); spacyr (Benoit & Matsuo, 2017); igraph (Csardi & Nepusz, 2006); lme4 (Bates, Mächler, Bolker, & Walker, 2015); vars (Pfaff, 2008); CORElearn (Robnik-Sikonja & Savicky, 2018); pandas (McKinney, 2012); numpy (van der Walt, Colbert, & Varoquaux, 2011); scipy (Jones, Oliphant, & Peterson, 2001); lxml (Behnel, Faassen, & Bicking, 2016); nltk (Bird, Klein, & Loper, 2009); spacy (Honnibal & Montani, 2019); statsmodels (Seabold & Perktold, 2010); scikit-learn (Pedregosa et al., 2011).

There is no one-size-fits-all approach to building a workflow, and several resources offer practical advice for those who want to get started. These resources range from freely available tutorials (e.g., Healy, 2018; Trilling, 2018) to books (e.g., Gandrud, 2015; Silge & Robinson, 2016).

For data cleaning and processing, it is strongly recommended to use a data analysis framework such as tidyverse or data.table for R and pandas for Python. In Python, it is also useful to look at the low-level packages numpy and scipy that undergird pandas. For scraping, both R and Python offer powerful libraries for retrieving data (httr, requests) and parsing Web pages (rvest, lxml), in addition to many special-purpose packages for dealing with, for example, the Twitter or Facebook APIs.

For text analysis, packages such as quanteda (for R) and nltk (for Python) are a good starting point. The Python spacy package, accessible in R through spacyr, offers excellent support for natural language processing in multiple languages. In both R and Python, igraph offers support for (social) network analysis.

Traditional statistics are mostly built in to R, although packages such as lme4 and VARS offer additional capabilities for multilevel and time series modeling, respectively. In Python, the statsmodels package offers most of the needed statistics. For machine learning, the Python scikit-learn platform is a great starting point, while for R, CORElearn is a good place to start. In general, when looking for R packages, it is often helpful to look up the relevant task view, which lists frequently used packages for specific topics or methods.[2]

Finally, it can be useful to consider a literate programming environment such as RMarkdown or Jupyter Notebooks. Literate programming allows the user to weave code and normal text in a single document. The tool then runs the code and integrates the results in the final document. This tool is helpful for creating tutorials and presentations, because the user can be sure that the example code works. It can also be used to create a fully reproducible paper or results section because it ensures that the provided code produced the presented results.

Conclusion

The computational analysis of the digital traces of communication and social behavior has the potential to provide an unprecedented boost to the field of communication science. To achieve this potential, our research should not just be reproducible but also replicable. Other researchers need to be able to conduct similar research in a similar setting with (hopefully) similar findings. However, for computational research to fulfill its promise, we need to move beyond this toward reusability, ensuring that data and code can be recombined, modified, and adapted in a process of open science.

To this end, we have proposed a practical guide toward reusability for researchers that stresses open sharing of data and code from data gathering to publication. In particular, data should be shared in a FAIR (findable, accessible, interoperable, reusable) way, and the provenance of the data should be published

[2] See, for example, the task list for machine learning by Hothorn (2019).

together with the data—ideally in the form of the code that was used to gather and process the data. The code that was used to analyze and model the data should be available as open source and should be structured in a modular fashion. Results should be published so the link between the presented findings and the data and analyses are clear, preferably using some form of literate programming. Finally, potentially reusable software should be distributed such that it can easily be found, used, and interpreted.

Who Can Be Against Reusability?

Our main argument is that the field of CCR will significantly benefit from fostering a culture of reusability. Moreover, we argue that individual researchers will profit from adapting reusability practices as outlined above. However, one could argue that taking these steps also puts a burden on the individual researcher. Klein and colleagues (2018) describe four possible obstacles: the risk of (unintentionally) violating people's privacy, the risk of being scooped, the risk of errors being revealed, and the time-consuming nature of the approach. In line with Klein and associates (2018), we see the risk of being scooped and the risk of errors as lesser hurdles. The former can be circumvented with embargos and is offset by increased visibility, and our field is not so crowded that many researchers are in competition to be first to publish a particular finding. Regarding the discovery of errors, this seems a feature rather than a bug, and it can be argued that an honest mistake is human and should not damage an author's reputation. In fact, as Muchlinski, Siroky, He, and Kocher (2019) demonstrate, when journals actively encourage replication and encourage the original authors to respond, error correction can—in the long run—become normal, and "well-established norms about what to do when postpublication replications identify errors" (p. 113) can emerge.

A more serious problem is the legal and ethical risks involved in sharing data. When sharing proprietary data, researchers and their institutions can be exposed to legal risks. Asking respondents to share their digital traces through, for example, a plug-in, we might even expose the respondent to legal risk by breaching the terms of use. Sharing personal data also exposes researchers to the dilemma between privacy and openness. In particular, the General Data Protection Regulation requires following a principle of "data minimization," which means not storing more data than strictly necessary and not storing data longer than necessary—a notion that is directly at odds with preserving data for secondary analyses. At the same time, the GDPR's explication of the rights of data subjects to access, rectify, and even erase their data is directly at odds with the goal of reproducibility. As we argue below, many of these risks will require institutional action to overcome. Fortunately, the legal and physical infrastructure to deal with these risks as an individual researcher has markedly improved, and it is mostly important to be aware of these risks and make sure to use adequate data management plans and coordinate data collection with the appropriate institutional bodies, such as the ethical committee or Institutional Review Board.

Finally, adopting the reusability practices suggested in this article can cost a researcher time that could otherwise be spent on conducting actual research. We agree with Klein and colleagues (2018), however, that engaging in data and code sharing will pay off for the individual researcher, because it makes it less time-consuming to revisit old projects. Moreover, increasing numbers of journals are requiring the submission of data and code to at least reproduce the presented results, and adopting reusability practices from the start of a research project can actually save time over having to gather, clean, and document the data and code after a submission is accepted.

A Strategic Road Map for the Field

Unfortunately, not all steps toward reusability can be made by individual researchers, and some of the challenges discussed here can only be met at the institutional level. Especially the challenges related to proprietary data and proper incentives for producing reusable code and data will require collective action as a field.

First, we need to define good standards for using, archiving, and sharing data protected by copyright, terms-of-use, and other legal restrictions. In the current situation, many tasks that researchers need to do to conduct good research—such as scraping websites, archiving the data, and sharing the data for reuse—are potentially illegal in some jurisdictions, and individual researchers and institutions make very different judgments about how to deal with this potential liability, from ignoring all rules and forging ahead with the research to making research almost impossible by carefully avoiding all liability. In the end, neither extreme will help, and the legal uncertainty makes it difficult to set up larger collaborations to gather and analyze large data sets. Creating guidelines for dealing with such data will make it clear what can and should be done to maximize the academic value without incurring legal liabilities. Having such shared guidelines will also make it easier to convince both institutions and the owners of the data (such as news publishers) of our good intentions and practices.

Second, we should make a concrete effort to explain to both governments and companies that the data we need to understand social behavior should be available to researchers and that we cannot let the data owners dictate the terms of their use. We must convince external parties, government, companies, and the public at large that such data are safe in our hands and that the greater good will benefit from academic research based on these data. For instance, a recent call by Bruns (2018) denounces the tightening of Facebook access to all but a select group of "privileged" scholars, and he calls for transparent and shareable access to social media data for researchers (see also Freelon, 2018). This is a good start, and as a field we need to continue pushing for this at all possible occasions. To this end, it is vital that we make it clear how academic research safeguards the data and interest of the people involved. Hence, we need to have clear and strictly enforced ethical guidelines and make sure that we have clear guidelines on how to store the data and how and under what conditions the data can be shared. To accomplish the last item, it can be beneficial to develop a shared academic data license and verifiable standards and procedures for dealing with data. If we can convince both the public and the data-owning companies that their data are safe with us and that we threaten neither their privacy nor their business model, it will be a lot easier to convince them to allow us to access this data.

Finally, the time-consuming nature of the practices we advocate in this article also needs to be addressed beyond the individual level. Although we believe reusability practices are beneficial for individual researchers, they require a degree of long-term thinking that might not be feasible to researchers on short-term contracts. Moreover, the steps needed to release software or data in a reusable way that goes beyond the requirements to reproduce a specific paper can take so much time that it will not be offset by future time-efficiency gains. We therefore strongly argue that this time investment needs to be made worthwhile and incentivized institutionally in the form of seeing data and tool contributions as stand-alone scientific contributions for the purpose of hiring, funding, or tenure decisions. Partly this needs to be done through

the relatively slow path of changing institutional guidelines and procedures, but we can also contribute directly by voicing our standards and criteria whenever we are in a position to judge our colleagues and their work. In our institutional roles as editors, reviewers, and committee members, we should do more to appreciate code and data as output indicators, especially if sufficient effort is made to make sure the code and data are reusable as described above.

Reusable Practices for Open Computational Communication Science?

Computational communication science offers the opportunity to increase the scope and accelerate the pace of the study of human communication. Increasing the reusability of data and code will be key to living up to this promise by improving the reproducibility of our research. Moreover, it will allow researchers to more easily join efforts toward producing the large data sets and sophisticated tools needed for CCS. By adopting the practical steps outlined in this article from the start of a project, individual researchers can produce more reusable data and code without investing undue extra effort. Overcoming the deeper challenges of dealing with proprietary and privacy-sensitive data and properly incentivizing data and tool sharing, however, will require more concerted institutional action. We hope that the arguments provided in this article will persuade both individuals and institutions to work toward these goals and thus contribute to a more open computational communication science.

References

Alvarez, R. M. (Ed.). (2016). *Computational social science: Discovery and prediction.* New York, NY: Cambridge University Press.

American Political Science Association. (2012). *APSA guide to professional ethics in political science* (2nd ed.). Washington, DC: Author. Retrieved from https://www.apsanet.org/TEACHING/Ethics

Bar-Sinai, M., Sweeney, L., & Crosas, M. (2016). Datatags, data handling policy spaces and the tags language. In *2016 IEEE Security and Privacy Workshops (SPW)* (pp. 1–8). Piscataway, NJ: Institute of Electrical and Electronics Engineers. doi:10.1109/SPW.2016.11

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. doi:10.18637/jss.v067.i01

Behnel, S., Faassen, M., & Bicking, I. (2016). lxml—XML and HTML with Python [Computer software manual]. Retrieved from https://lxml.de/

Benoit, K., & Matsuo, A. (2017). spacyr: Wrapper to the "spacy" "NLP" library (R package version 0.9.0) [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=spacyr

Benoit, K., Watanabe, K., Nulty, P., Obeng, A., Wang, H., Lauderdale, B., & Lowe, W. (2017). quanteda: Quantitative analysis of textual data (R package version 0.99) [Computer software manual]. Retrieved from http://quanteda.io

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit.* Sebastopol, CA: O'Reilly.

Bowman, N. D., & Keene, J. R. (2018). A layered framework for considering open science practices. *Communication Research Reports*, *35*(4), 1–10. doi:10.1080/08824096.2018.1513273

boyd, d., & Crawford, K. (2012). Critical questions for big data. *Information, Communication & Society*, *15*(5), 662–679. doi:10.1080/1369118X.2012.678878

Bruns, A. (2018, April 25). Facebook shuts the gate after the horse has bolted, and hurts real research in the process. *Internet Policy Review*. Retrieved from https://policyreview.info/articles/news/facebook-shuts-gate-after-horse-has-bolted-and-hurts-real-research-process/786

Cappella, J. N. (2017). Vectors into the future of mass and interpersonal communication research: Big data, social media, and computational social science. *Human Communication Research*, *43*(4), 545–558. doi:10.1111/hcre.12114

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems* (pp. 288–296). Red Hook, NY: Curran Associates.

Chavan, V., & Penev, L. (2011). The data paper: A mechanism to incentivize data publishing in biodiversity science. *BMC Bioinformatics*, *12*(15), S2. doi:10.1186/1471-2105-12-S15-S2

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, *Complex Systems*, 1695. Retrieved from http://igraph.sf.net

Dowle, M., & Srinivasan, A. (2017). data.table: Extension of "data.frame" [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=data.table

Ekbia, H., Mattioli, M., Kouper, I., Arave, G., Ghazinejad, A., Bowman, T., . . . Sugimoto, C. R. (2015). Big data, bigger dilemmas: A critical review. *Journal of the Association for Information Science and Technology*, *66*(8), 1523–1545.

FORCE11. (2016). The FAIR data principles. Retrieved from https://www.force11.org/group/fairgroup/fairprinciples

Freelon, D. (2018). Computational research in the post-API age. *Political Communication*, *35*(4), 665–668. doi:10.1080/10584609.2018.1477506

Gandrud, C. (2015). *Reproducible research with R and RStudio* (2nd ed.). Boca Raton, FL: CRC Press.

Gil, Y., Groth, P., Ratnakar, V., & Fritz, C. (2009). Expressive reusable workflow templates. In 2009 *Fifth IEEE International Conference on e-Science* (pp. 344–351). Piscataway, NJ: Institute of Electrical and Electronics Engineers. doi:10.1109/e-Science.2009.55

González-Bailón, S. (2017). *Decoding the social world: Data science and the unintended consequences of communication.* Cambridge, MA: MIT Press.

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. doi:10.1093/pan/mps028

Günther, E., Trilling, D., & Van de Velde, B. (2018). But how do we store it? Data architecture in the social-scientific process. In C. M. Stuetzer, M. Welker, & M. Egger (Eds.), *Computational social science in the age of big data* (pp. 161–187). Cologne, Germany: Herbert von Halem.

Hargittai, E. (2015). Is bigger always better? Potential biases of big data derived from social network sites. *Annals of the American Academy of Political and Social Science*, *659*(1), 63–76. doi:10.1177/0002716215570866

Healy, K. (2018, April 28). *The plain person's guide to plain text social science.* Retrieved from http://plain-text.co/plain-person-text.pdf

Honnibal, M., & Montani, I. (2019). *spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing*. Retrieved from https://spacy.io

Hothorn, T. (2019). CRAN task view: Machine learning and statistical learning. Retrieved from https://cran.r-project.org/web/views/MachineLearning.html

Hutton, C., Wagener, T., Freer, J., Han, D., Duffy, C., & Arheimer, B. (2016). Most computational hydrology is not reproducible, so is it really science? *Water Resources Research*, *52*(10), 7548–7555. doi:10.1002/2016WR019285

Jones, E., Oliphant, T., Peterson, P., et al. *[sic].* (2001). *SciPy: Open source scientific tools for Python.* Retrieved from http://www.scipy.org/

Kitchin, R. (2014). Big data, new epistemologies and paradigm shifts. *Big Data and Society*, *1*(1), 1–12. doi:10.1177/2053951714528481

Klein, O., Hardwicke, T. E., Aust, F., Breuer, J., Danielsson, H., Hofelich Mohr, A., . . . Frank, M. C. (2018). A practical guide for transparency in psychological science. *Collabra: Psychology*, *4*. doi:10.1525/collabra.158

Kosinski, M., Stillwell, D., & Graepel, T. (2013). Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, *110*(15), 5802–5805. doi:10.1073/pnas.1218772110

Lazer, D., Pentland, A. S., Adamic, L., Aral, S., Barabasi, A. L., Brewer, D., . . . Alstyne, M. V. (2009). Computational social science. *Science*, *323*(5915), 721–723. doi:10.1126/science.1167742

Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., . . . Zhao, Y. (2006). Scientific workflow management and the Kepler system. *Concurrency and Computation: Practice and Experience*, *18*(10), 1039–1065. doi:10.1002/cpe.994

Marwick, B. (2019). rrtools: Creates a reproducible research compendium (R package version 0.1.0) [Computer software manual]. Retrieved from https://github.com/benmarwick/rrtools

Marwick, B., Boettiger, C., & Mullen, L. (2018). Packaging data analytical work reproducibly using R (and friends). *American Statistician*, *72*(1), 80–88. doi:10.1080/00031305.2017.1375986

McKinney, W. (2012). *Python for data analysis*. Sebastopol, CA: O'Reilly.

Miguel, E., Camerer, C., Casey, K., Cohen, J., Esterling, K. M., Gerber, A., . . . Van der Laan, M. (2014). Promoting transparency in social science research. *Science*, *343*(6166), 30–31. doi:10.1126/science.1245317

Muchlinski, D. A., Siroky, D., He, J., & Kocher, M. A. (2019). Seeing the forest through the trees. *Political Analysis*, *27*(1), 111–113. doi:10.1017/pan.2018.45

Munafò, M. R., Nosek, B. A., Bishop, D. V., Button, K. S., Chambers, C. D., Du Sert, N. P., . . . Ioannidis, J. P. (2017). A manifesto for reproducible science. *Nature Human Behaviour*, *1*(1), 21. doi:10.1038/s41562-016-0021

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., . . . Yarkoni, T. (2015). Promoting an open research culture. *Science*, *348*(6242), 1422–1425. doi:10.1126/science.aab2374

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pfaff, B. (2008). VAR, SVAR and SVEC models: Implementation within R package vars. *Journal of Statistical Software*, *27*(4). doi:10.18637/jss.v027.i04

Piccolo, S. R., & Frampton, M. B. (2016). Tools and techniques for computational reproducibility. *GigaScience*, *5*(1). doi:10.1186/s13742-016-0135-4

Robnik-Sikonja, M., & Savicky, P. (2018). CORElearn: Classification, regression and feature evaluation (R package version 1.53.1) [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=CORElearn

Salganik, M. J. (2017). *Bit by bit: Social research in the digital age.* Princeton, NJ: Princeton University Press.

Seabold, S., & Perktold, J. (2010, June). *Statsmodels: Econometric and statistical modeling with Python.* In S. van der Walt & J. Millman (Eds.), *Proceedings of the Ninth Python in Science Conference* (Scipy 2010), 57–61. Austin, TX.

Shah, D. V., Cappella, J. N., & Neuman, W. R. (2015). Big data, digital media, and computational social science: Possibilities and perils. *Annals of the American Academy of Political and Social Science*, *659*(1), 6–13. doi:10.1177/0002716215572084

Silge, J., & Robinson, D. (2016). tidytext: Text mining and analysis using tidy data principles in R. *Journal of Open Source Software*, *1*(3). doi:10.21105/joss.00037

Stockemer, D., Koehler, S., & Lenz, T. (2018). Data access, transparency, and replication: New insights from the political behavior literature. *PS: Political Science and Politics*, 1–5. doi:10.1017/S1049096518000926

Trilling, D. (2017). Big data, analysis of. *International Encyclopedia of Communication Research Methods*. Hoboken, NJ: John Wiley. doi:10.1002/9781118901731.iecrm0014

Trilling, D. (2018, January 21). Doing computational social science with Python: An introduction. *SSRN*. Retrieved from http://papers.ssrn.com/abstract=2737682

van Atteveldt, W. (2008). *Semantic network analysis: Techniques for extracting, representing, and querying media content*. Charleston, SC: BookSurge.

van Atteveldt, W., & Peng, T.-Q. (2018). When communication meets computation: Opportunities, challenges, and pitfalls in computational communication science. *Communication Methods and Measures*, *12*(2–3), 81–92. doi:10.1080/19312458.2018.1458084

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, *13*(2), 22–30. doi:10.1109/mcse.2011.37

Wallach, H. (2016). Computational social science: Towards a collaborative future. In R. M. Alvarez (Ed.), *Computational social science: Discovery and prediction* (pp. 307–316). New York, NY: Cambridge University Press.

Wickham, H. (2015). rvest: Easily harvest (scrape) web pages. Retrieved from https://rvest.tidyverse.org/

Wickham, H. (2018). httr: Tools for working with URLs and HTTP (R package version 1.4.0) [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=httr

Wickham, H., & Grolemund, G. (2017). *R for data science: Import, tidy, transform, visualize, and model data.* Sebastopol, CA: O'Reilly.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., . . . Mons, B. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, *3*. doi:10.1038/sdata.2016.18

Zeng, J., Ruan, G., Crowell, A., Prakash, A., & Plale, B. (2014). Cloud computing data capsules for non-consumptive use of texts. In *Proceedings of the 5th ACM Workshop on Scientific Cloud Computing* (pp. 9–16). New York, NY: Association for Computing Machinery. doi:10.1145/2608029.2608031