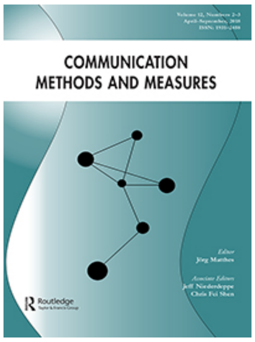

Routledge
Taylor & Francis Group

Communication Methods and Measures

ISSN: 1931-2458 (Print) 1931-2466 (Online) Journal homepage: https://www.tandfonline.com/loi/hcms20

Scaling up Content Analysis

Damian Trilling & Jeroen G. F. Jonkman

To cite this article: Damian Trilling & Jeroen G. F. Jonkman (2018) Scaling up Content Analysis, Communication Methods and Measures, 12:2-3, 158-174, DOI: 10.1080/19312458.2018.1447655

To link to this article: https://doi.org/10.1080/19312458.2018.1447655

COMMUNICATION METHODS AND MEASURES
2018, VOL. 12, NOS. 2–3, 158–174
https://doi.org/10.1080/19312458.2018.1447655

OPEN ACCESS

Scaling up Content Analysis

Damian Trilling and Jeroen G. F. Jonkman

Department of Communication Science Amsterdam School of Communication Research, University of Amsterdam, Amsterdam, The Netherlands.

ABSTRACT

Employing a number of different standalone programs is a prevalent approach among communication scholars who use computational methods to analyze media content. For instance, a researcher might use a specific program or a paid service to scrape some content from the Web, then use another program to process the resulting data, and finally conduct statistical analysis or produce some visualizations in yet another program. This makes it hard to build reproducible workflows, and even harder to build on the work of earlier studies. To improve this situation, we propose and discuss four criteria that a framework for automated content analysis should fulfill: scalability, free and open source, adaptability, and accessibility via multiple interfaces. We also describe how to put these considerations into practice, discuss their feasibility, and point toward future developments.

Introduction

Manual content analysis is still one of the core methods used in Communication Science (e.g., Lacy, Watson, Riffe, & Lovejoy, 2015). But in our digitized media environment, datasets grow larger and larger, which is why automated content analysis (ACA) has gained importance and popularity (Boumans & Trilling, 2016; Grimmer & Stewart, 2013; Günther & Quandt, 2016; Jacobi, Van Atteveldt, & Welbers, 2016; Scharkow, 2011). It has been used in, for instance, studies about sources and topics in news (e.g., Burscher, Vliegenthart, & De Vreese, 2015; Scharkow, 2011; Sjøvaag & Stavelin, 2012) or agenda setting and framing (e.g., Burscher, Odijk, Vliegenthart, de Rijke, & de Vreese, 2014; Russell Neuman, Guggenheim, Mo Jang, & Bae, 2014; Tsur, Calacci, & Lazer, 2015). In this article, rather than introducing a new technique or tool, we reflect on the current state of scaling up content analysis and present a set of guidelines to further advance the field.

From time to time, a discipline needs to pause for a moment and reflect on its own theories and methods. For instance, in a 1983 special issue of the *Journal of Communication* on the "Ferment in the field",[1] Gerbner (1983) argued that the discipline of communication science "requires an intellectual domain, a body of theories and approaches that fit its subject matter [...]" (p. 355). Given the enormous changes in the possibilities of one of the most central methods of our field, it is worth to pause again for a moment and reflect on how approaches and best practices could look like. While we are somewhat reluctant to use the often abused buzzword Big Data, the emerging field of *Computational Social Science* (Cioffi-Revilla, 2014; Lazer et al., 2009)—or, in the words of Shah, Cappella, and Neuman (2015), *Computational Communication Science*—deals with research questions and data sets that require re-thinking traditional approaches to content analysis.

CONTACT Damian Trilling d.c.trilling@uva.nl Department of Communication Science Amsterdam School of Communication Research, University of Amsterdam, Amsterdam, The Netherlands.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/hcms.

[1] Comparable special issues were published in 1993, 2008, and 2018.

Recently, several scholars have published thoughtful pieces on this development. For instance, Kitchin (2014a) has written on the epistemological and paradigm shifts that come along with the use of Big Data in the social sciences, and Freelon (2014b) has reflected on how the analysis of digital trace data can be fruitfully incorporated into the discipline of communication science. In a special issue on "Toward Computational Social Science: Big Data in Digital Environments" of *The ANNALS* of the American Academy of Political and Social Science, several authors give specific advice on how to incorporate recent methodological and computational developments into our discipline. For example, Hindman (2015) discusses how machine learning-related approaches can be used within communication science, and Zamith and Lewis (2015) discuss the possibility of hybrid forms of content analysis. Furthermore, in the afterword of a special issue of the *Journal of Communication* on "Big Data in Communication Research", Parks (2014) concludes that we face "significant challenges [...] as we move into the era of Big Data" (p. 360).

With this article, we want to contribute to this discussion of where the field is heading. Our field is at crossroads: As computational methods are gaining ground, there is a pressing need to establish standards of *how* these techniques should be employed and implemented. Going beyond suggestions for specific standards, programs, or environments, we set out to develop a series of criteria that a framework for scaled-up ACA needs to fulfill. By framework, we mean a coherent combination of research practices, software packages, and hardware solutions, which together allow to conduct scaled-up ACA. We explicitly do not want to come up with "yet another standard",[2] but rather want to present criteria that may be worth considering when our discipline is working on scaling up content analysis.

While also manual content analyses nowadays make use of computers (for instance, to facilitate data entry; e.g., Lewis, Zamith, & Hermida, 2013), we only consider those forms of content analysis "automated" where the coding itself is not performed manually; vice versa, also automated content analyses need human insight and human decisions. A very basic form of ACA entails the coding the occurance of some terms in a dataset, which can be accomplished in standard and easy-to-use software like Stata, SPSS, or Excel (Boumans & Trilling, 2016; Krippendorff, 2004). As the actual counting is not performed by hand and the texts themselves are not read by the researchers, even such simple approaches can be classified as ACA. Nevertheless, they are not "scaled up": When applied to only a couple of thousands of items, such an analysis does not require the researcher to think about the necessary infrastructure. Neither hardware nor software requirements exceed what any standard laptop offers. But this changes drastically once datasets grow larger—or when datasets do not consist of plain text, but include images, video, or other types of data.

The prevalent approach among many communication scholars who use computational methods to analyze large collections of texts has for a long time been to employ a number of different standalone programs—and this is also how common content analysis textbooks present computer-aided methods (e.g., Krippendorff, 2004).[3] While this is somewhat understandable, as it is easier to get financing for a project with a clearly defined scope than for the development and the maintenance of a more general framework, in the long run, doing the latter will save resources, allow to answer new research questions, and facilitate collaboration. Such a framework can and probably should include the use of environments like R or Python, which already include many libaries and therefore allow for unifying the workflow (see, e.g., Freelon, 2014a), but goes beyond this. For instance, it would need to also include other components such as a database backend.

Recently, the use and development of more sustainable systems gained momentum within communication science. Examples that are clearly aiming at providing a framework that is not tied to a single project include the AmCAT framework developed at Vrije Universiteit Amsterdam

[2] We would like to thank an anonymous reviewer for pointing us to the comic strip available at https://xkcd.com/927/, which claims that an attempt to unify 14 competing standards will result in 15 competing standards.

[3] This situation has improved, though, since more and more social scientists use environments like R or Python that allow for integrated workflows. Nevertheless, the large market for commercial tools like Provalis Wordstat or standalone-tools developed by individual scientists (e.g., http://leydesdorff.net/) illustrate that the use of standalone programs is still considerable.

(Van Atteveldt, 2008) and used at several other departments as well; the xTAS framework, which is geared more towards computer science (De Rooij, Vishneuski, & De Rijke, 2012); or the Leipzig Corpus Miner (Niekler, Wiedemann, & Heyer, 2014). In this paper, we want to go beyond discussing the merits and shortcomings of individual frameworks, but aim at bringing this discussion to a higher level, on which we conceptualize guidelines for designing such frameworks.

Unfortunately, such frameworks are not in widespread use across the discipline yet. What is particularly problematic is that—without such a framework—a lot of work is unnecessarily repeated, which happens too often. For example, if texts from a certain source have already been collected, or if such texts have already been pre-processed or coded (annotated) in some way or another, it should be avoided to do this tedious work again in a similar project. This should be easy with digitally available data, but the use of different, partly proprietary programs, makes it difficult to cooperate and replicate or enhance existing research. While it is a well-known problem that researchers often are too reluctant to share data; making it easier to actually share data by integrating data sharing facilities into a framework of ACA might help alleviate the problem.

Although automated content analysis techniques have been employed decades ago (as an illustration, see, e.g., a study analyzing word co-occurrences by Leydesdorff & Zaal, 1988), they only recently became widely adapted in communication science (see, e.g., Boumans & Trilling, 2016). However, also today, many of automated content analyses still follow the same approach of such early studies, in that they rely on standalone tools to solve a problem at hand. For example, a wide range of programs exist to handle tasks like processing text files, counting word frequencies, or visualizing co-occurrences (see, e.g, the large amount of different tools suggested Krippendorff, 2004). Similarily, the Digital Methods Initiative developed a set of excellent tools[4] for conducting some very specific analyses tailored to online content. However, using a number of different standalone programs and services that cannot be chained together without manual interventions, is neither transparent nor scalable.

Partly, these problems can be solved by using a framework like R, which is very extensible and therefore allows researchers to do many of these steps within one environment. But even though this is a very good step toward more reproducible research, having standalone R installations on researchers' computers does not completely solve the problem. In particular, one would at least need some common database backend to share the data.

We therefore started thinking about implementing a framework that can be used across research projects and allows re-running models, extending them, and avoid redundant work in data collection and preprocessing. We argue that such a framework should incorporate state-of-the art libraries (e.g., scikit-learn, Pedregosa et al., 2011; or gensim, Řehůřek & Sojka, 2010) and allow people who do have some programming knowledge to extend it, but at the same time be usable for others as well. While thinking about the design of such a system, we quickly realized that the scale of the system goes beyond what communication researchers usually can handle on their laptop or desktop computer. But, more importantly, even if it is technically possible to run the analysis on a local computer, it makes sense to move to a larger scale, in order to make better long-term use of the data.

In this article, we therefore (1) ask what the criteria are that a framework for scaled-up content analysis has to fulfill, (2) describe how to put these considerations into practice, and (3) point toward future developments.

Scaling up content analysis

The decision for scaling up content analysis is a decision that frequently arises from the need to keep pace with the "data revolution" (Kitchin, 2014b). We have more data and different data at our disposal, and these data are available in digital formats. From a technical point of view, this can lead to three bottlenecks. We will first discuss the relevance of these points before moving on toward developing a more encompassing and broader set of criteria for scaled-up content analysis.

[4] They are available at https://wiki.digitalmethods.net/Dmi/ToolDatabase.

(1) A lack of storage capacity, i.e., the dataset is too large to store on the own harddrive.
(2) A lack of memory, i.e., the part of the dataset needed for some calculation is too large to be loaded.
(3) A lack of computing power, i.e., the calculation would take prohibitively long.

First, the lack of storage capacity usually is a minor issue: It is inconvenient if a dataset is too large to store on the researcher's laptop, but in an age where external hard drives of several terabytes are affordable and offer more space than even several decades of newspaper data (or, e.g., comprehensive sets of press releases, parlamentary speeches, etc.) require, this would not prevent us from conducting a content analysis. However, there are cases where storage is a problem. For example, Yahoo has released a Yahoo News dataset, which, although it only contains four months of data, has an uncompressed size of 13.5 TB, which is larger than the size of external hard drives available on the consumer market.

Second, the lack of memory can be a much more important limiting factor. For a lot of analyses, it is necessary to construct a matrix of, for instance, word frequencies. Such a matrix can easily outgrow the computer's memory, effectively limiting the number of documents or the number of words that can be taken into account. But problems can start much earlier: A data set of 8 GB size (which is not too uncommon) cannot even be loaded into the 8 GB memory that a laptop might have, so we have to let go of the idea of "opening" a file and instead have to somehow process it in parts.

Third, the lack of computing power deserves attention: It might be acceptable to wait even some days for the result of an analysis. But it is usually not acceptable to block the researcher's own computer for such a long time, especially if additional requirements like permanent high-speed internet connection, have to be fulfilled. And also if a dedicated computer is available, even longer calculation times might not be acceptable. To illustrate the scope of the problem, let us consider a study that aims at comparing different texts. For example, a researcher might want to find out whether different articles are based on the same original material (e.g., Boumans, 2016; Welbers, van Atteveldt, Kleinnijenhuis, & Ruigrok, 2016). However, the number of neccessary comparisons increases exponentially. If the dataset comprises 100 articles, $100 \cdot 100 = 10,000$ comparisons have to be made; with a still not too high number of 10, 000 articles, this increases to 100, 000, 000 comparisons, and quickly, we arrive in regions that become problematic both in terms of memory and computing time.

As the examples described above illustrate, a naïve approach (i.e., load all data into memory, compare everything with everything) is perfectly fine with small datasets, but once we scale up, problems arise. Thus, we need to carefully think about how to build a framework that deals with such tasks in a smarter way.

One might object that the advent of cloud-computing services has turned these challenges from unsurmountable ones to ones that can be tackled. This is indeed the case, but in fact it actually highlights the need for the systematic planning of a research infrastructure. After all, using such services come with certain requirements—like, in general, some programming knowledge, and the selection of tools and approaches that can be efficiently run on such a cloud-computing platform. In addition, as a framework for scaled-up content analysis should also be sustainable (for instance, reusable for later projects), the task of scaling up content analysis is not one of mere availability of storage, memory, and computational power.

Using our own research practice and that of our colleagues to inform our choices, we see four essential requirements.

(1) The framework should be usable on a laptop, but should be designed in a way that it can be run on a powerful server or even a cluster of servers to analyze millions of documents (*scalability*).

(2) The framework should not depend on any commercial software and run on all major operating systems, which, in fact, is also necessary to satisfy the first criterion. In practice, that means that it needs to be *free and open source*.

(3) The framework should be flexible and programmed in a way that facilitates users in adopting it to their own needs and to use it for collaboration on a wide range of projects (*adaptability*). The second criterion is a precondition for this.

(4) The framework should have a powerful database engine on the background which gives the advanced user full control; but at the same time, there must be a easy-to-use interface for the inexperienced user (*multiple interfaces*).

We discuss these four criteria one by one in the following subsections.

Scalability

Traditional content analyses typically deal with a number of cases in the order of magnitude of thousands. In some rare exceptions, tens of thousands are reached. But even in a massive study of news coverage of the 2009 EP elections in all 27 EU member states (Schuck, Xezonakis, Elenbaas, Banducci, & De Vreese, 2011), "only" $N = 52,009$ news items were (manually) analyzed. However, larger and larger collections of texts become digitally available and, as computing is cheap while human coders are expensive, it does not make sense to draw a sample of texts to be analyzed. This has been famously dubbed as the N = all approach of Big Data analysis (Mayer-Schönberger & Cukier, 2013). In the words of Kitchin (2014a), "Big Data is characterized by being generated continuously, seeking to be exhaustive and fine-grained in scope, and flexible and scalable in its production" (p. 2).[5] Consequently, this means that methods for analyzing such data have to be scalable as well.

Scalability and the efficiency of the implementation of an analysis could be neglected when the amount of data was limited. But this has changed, and it is argued that "[s]ocial research has to embrace more efficient and scalable methods if it is to use online data to improve current models and theories" (Gonzalez-Bailon & Paltoglou, 2015, p. 106). In fact, one does not need to analyze really big Big Data to run into scalability problems.

For example, in a study in which $N = 6,142$ agency releases articles had to be linked to $N = 22,928$ news articles, Boumans (2016) used a specific custom-written program to do so. However, when he wanted to conduct a similar analysis on sets of about $N = 100{,}000$ and $N = 250{,}000$ items, he found that re-using the software from the smaller study was unfeasible, both in terms of memory and computing time needed. While on a small dataset, the difference between the old program and the new, more efficient program was irrelevant (it does not really matter whether it takes a few minutes or some hours to arrive at the result), this difference in efficiency translated to few hours vs. several weeks on the larger dataset. Such analyses that focus on research questions asking about the overlap between collections of texts are a prime example that can only be answered in a scaled-up framework: human coding is inherently impossible, as remembering the content of so many articles to find it back in other articles exceeds the capacity of the human brain; and small-scale standalone programs to link similar strings (e.g., Schnell, Bachteler, & Reiher, 2005) are designed for other purposes and simply cannot deal with such large datasets.

To ensure scalability, one thus has to consider the efficiency of the algorithms used. For example, communication scholars conducting large-scale automated content analysis are often interested in co-occurrences of words in texts, such as media articles, internet posts, and institutional communication (e.g., Hellsten, Dawson, & Leydesdorff, 2010; Jonkman & Verhoeven, 2013; Leydesdorff &

[5] While we have argued above that *Big Data* is not the most appropriate term for the type of data most social scientists work with, the general argument here remains true for smaller, but still very large datasets: There is little reason to draw a sample if enough computing power is available.

Welbers, 2011; Van der Meer, Verhoeven, Beentjes, & Vliegenthart, 2014). When comparing words in hundreds of thousands of texts, most words do actually *not* occur in a given text, and most texts actually are *not* linked to other texts (e.g., Jonkman & Verhoeven, 2013).

Consequently, a matrix of such data structures contains a lot of zeros. A much more efficient way of doing calculations based on these data is therefore to use a so-called sparse matrix, in which only the non-zero values are stored. Somewhat related, packages achieve scalability by using streaming techniques that do not load all data into memory before estimating a model, but update the model continuously while reading the data, as for example the topic modeling package by Řehůřek and Sojka (2010) does. More generally, whether a given algorithm is efficient or not does not matter in small datasets, but becomes increasingly important once we scale up content analysis.

A second consideration is the need for a modular architecture. By this, we mean that in a framework for scaled-up content analysis, it must be possible to have different parts of the framework located at different systems. For instance, the actual analysis does not necessarily need to be conducted on the same computer as where the data is stored. For example, to run the analyses in his dissertation, Van Atteveldt (2008) ran the AmCAT (Amsterdam Content Analysis Toolkit) infrastructure, which he developed, on two servers: a dedicated database server and a dedicated script server. Of course, one could also opt to install both on the same physical machine. Similarly, the xtas (eXtensible Text Analysis Suite) infrastructure (De Rooij et al., 2012) as well as the Leipzig Corpus Miner (Niekler et al., 2014) distinguish between workers, a web frontend, and a database. While all of this can be run locally on one computer, the framework is scalable in that it also allows of distributing tasks between a number of machines. Similarily, toolkits for analyzing social media messages rely on SQL database servers (e.g., DMI-TCAT, Borra & Rieder, 2014) or NoSQL database servers (e.g., smappPy, Social Media and Political Participation Lab at New York University, 2016), which may or may not be run on the same machine as where the analyses are conducted. Extending this line of reasoning, one could also argue that using MapReduce-frameworks like Hadoop can be of added value here. However, most datasets used in our discipline do not reach a size that would require this.[6]

Open source

From the above, it follows that we need to be able to run our framework on (multiple) real or virtual machines. Already from a purely financial and practical standpoint, it becomes clear that it is impractical to acquire and administer multiple licenses for each and every machine. But there are more reasons to strive for independence from closed-source and proprietary software.

Without a doubt, the scientific community has profited from the rise of open software and a culture of sharing source code free of charge (e.g., Günther & Quandt, 2016). This is not only true in financial terms, but also because it allows re-combining and improving existing code without having to reinvent the wheel all the time. In particular, Python and R, two languages used extensively in the data science community, but also in communication science research, are continuously extended by additional modules that users share with the community, which leads to a virtous circle of increasing popularity and improvement of the software. This popularity also means that many cutting-edge techniques are first implemented in such open-source frameworks, before they are used in proprietary packages. For instance, Günther and Quandt (2016) note that:

> "with open-source toolsets like Voyant Tools (voyant-tools.org) and vivid user communities for R and Python, there are many resources available for social scientists who aim to include automated text analysis methods into their projects. Using sophisticated tools and resources from disciplines such as computer science and computational linguistics, journalism scholars can gain insight into the constant information flow and make big data a regular feature in the scientific debate" (p. 86).

[6] For an example of an exception, see Lansdall-Welfare et al. (2017).

Examples for such open-source components that can be included in a content-analysis framework include highly popular modules like the Natural Language Toolkit NLTK (Bird, Loper, & Klein, 2009) or the Stanford Natural Language Processing tools (Manning et al., 2014), but also general-purpose libraries for efficient computation of, e.g., machine-learning problems (e.g., Pedregosa et al., 2011; Řehůřek & Sojka, 2010).

It is also important to consider the use of proprietary closed-source software from the perspective of research standards. First of all, if the source code is not open, the software is essentially a black box. For instance, if one uses a proprietary package to do a sentiment analysis, but the underlying algorithm is not public, this does not contribute much to scientific knowledge and errors might not be discovered (Broussard, 2016; Busch, 2014). In line with this argument, Heiberger and Riebling (2016) identify one big problem of contemporary social science research, namely "using a patchwork of different, specialized programs most of which are proprietary, thereby making it almost impossible to know their exact inner workings or definitions and resulting in an inflexible workflow" (p. 6).

In addition, it is impossible to reproduce the results when the software is not available any more. An open-source algorithm, on the other hand, can be re-implemented again, even for systems that do not even exist yet. In such a long-term perspective, the reliance on open-source solutions also mitigates the risk of vendor lock-in, i.e., the problem that once one has chosen for a specific (proprietary) system, it is hard or even impossible to switch to another one. For example, if a system would not store or at least is able to export its data in a widely recognized open format (e.g., CSV, XML, JSON), it might be hard or even impossible to transfer it to another system.

Moreover, if all parts of a content analysis framework are kept open source, results can be replicated by any researcher, regardless of their financial means.

Adaptability

The two aforementioned criteria (scalability and open-source) are related to a third criterion, which we refer to as adaptability. By this, we mean that the framework should be flexible enough to be adapted with a reasonable effort. For example, it should be possible to extend the framework by including new functions for analyses or new input filters for new types of content. Users with some technical knowledge should be able to make such changes, which—in general—is achieved by using open-source components and popular programming environments. In addition, also less technically savvy users should be able to tailor the analyses to their needs. For example, rather than just offering an option to remove stopwords, it should be possible to modify the stopword list. This also means that such options should not be hard-wired in the program itself, but rather be provided in the form of configuration files that can be easily changed by the users themselves. This makes it also possible to use tailored configuration files for each research project one runs. Communication scholars are increasingly interested in communication data entailing multiple character encodings. For instance, social media data (e.g., Twitter and Facebook) standardly includes characters such as emoticons, hashtags, and mixed languages. Adaptability therefore also means that there should not be any inherent barriers that prevent analyzing context from different languages and scripts. While the unicode standard, for example, has been in place for decades, there are still some tools in use that operate only on the basis of the limited set of 256 ASCII-characters. This means that each emoticon in a social media message, and each quotation in Hebrew or Arabic, cannot be handled. Even though most natural language processing techniques are only implemented for a few languages up until now, there should be the possibility to integrate support once such a resource becomes available.

Another important aspect of adaptability concerns the ability to integrate (by-)products of own research projects to use them for later projects. In practice, we often see that once a (Ph.D.-) project is finished, models and other resources that have been build disappear. For instance, a researcher might have trained some machine learning models, or created an extensive list of regular expressions, or any other asset. Losing information and knowledge is not only costly and inefficient in the long run, it may also pose severe problems in the case of longitudinal projects where several consecutive studies may build on each other. Hence, a flexible framework should make it easy to add resources, in order to enable others to re-use them.

Simple interface for beginners, but no limits for power users

At this point, one of the obstacles for wide-spread use of ACA is the lack of easy-to-use tools. While a lot of techniques are known, especially in fields like computer science, there are comparably few communication scientists who can apply these methods (for a discussion on this see Boumans & Trilling, 2016). This is actually a dilemma: While having an easy-to-use tool would enable more people to apply ACA methods, this is potentially at odds with the adaptability criterion the scalability criterion formulated above. Consequently, even though it may be tempting in the short term, ease-of-use must not be achieved at the expense of other crucial conditions for being able to scale up content analysis.

To solve this dilemma, we suggest that a framework for scaled-up content analysis should provide multiple ways of accessing it. This makes it accessible for different groups of researchers with different needs. For instance, one can think of a combination of the following elements:

- a web interface that provides point-and-click access to the most common analyses and allows exporting subsets of the data;
- a web interface that allows the use of a flexible query language;
- a command-line interface to allow scripting and batch processing of (complicated or time-consuming) analyses;
- an API to link the system to statistical packages like R; and
- direct access to the underlying database, bypassing the whole toolkit itself.

Such a design makes the infrastructure usable for research groups with diverse needs and different levels of technical understanding.

Scaling up in practice

Having established the criteria for scaling up content analysis, we would like to emphasize that the journey towards scaled-up content analysis is a continuum rather than a binary either/or question. We tried to systematize different dimensions of scalability in Figure 1.

One dimension of scaling up is scaling up the scope of the project – moving from one-off studies (P1 in the figure) via re-using data (P2) to a permanent, flexible framework for cross-project data collection and analysis (P3), as indicated by the upper x-axis. For instance, a number of projects at the University of Amsterdam joined forces to build an infrastructure as we will present in this chapter to create synergies and answer research questions reaching from an investigation of differences between online and offline news (Burggraaff & Trilling, 2017), via the predictors of news sharing (Trilling, Tolochko, & Burscher, 2017) to the characteristics of company news coverage (Jonkman, Trilling, Verhoeven, & Vliegenthart, 2016) and the influence of such coverage on stock rates (Strycharz, Strauss, & Trilling, 2018). All of these studies were conducted using a Python-based framework with a NoSQL database backend, making use of techniques ranging from regular-expression based keyword searches to supervised and unsupervised machine learning, and further analyzing the aquired data using regression models and time series analysis.

Collecting data not for a single project, but for re-use in various projects, goes hand in hand with scaling up the size of the dataset, as the lower x-axis in Figure 1 indicates. Let us illustrate the scale of such analyses with some more—admittedly rather arbitrary—examples. Manual content analyses (M1) usually comprise a couple of hundreds of items. An upper bound may be given by an example of a massive project with an exceptionally high number of more than 50,000 news items (Schuck et al., 2011), which involved dozens of annotators.

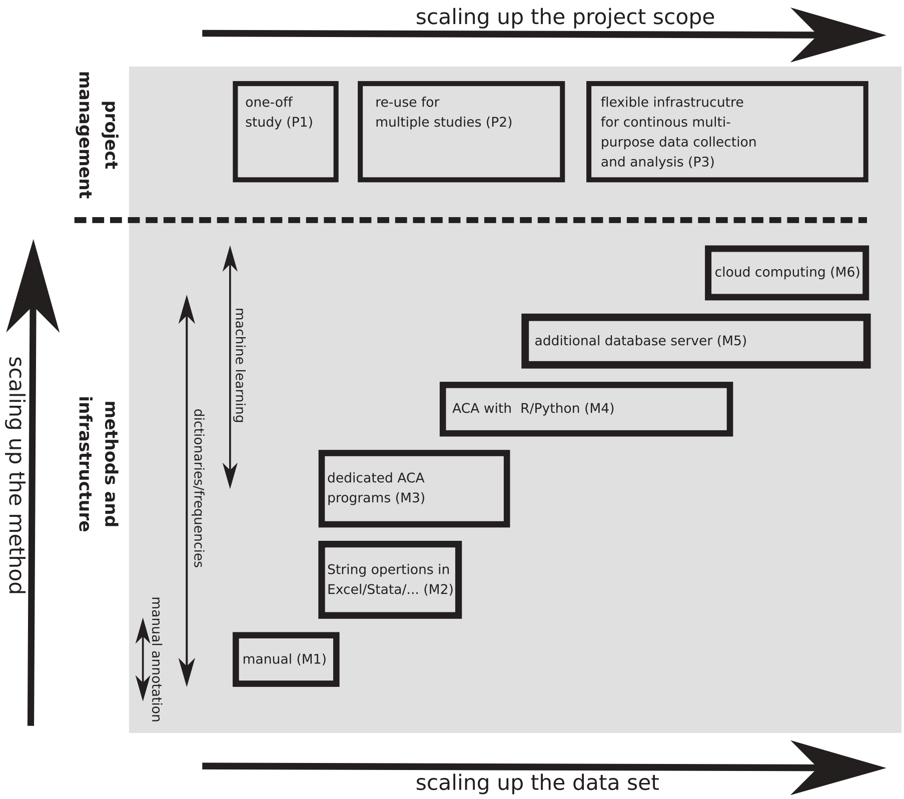
scaling up the project scope
project management
one-off study (P1)
re-use for multiple studies (P2)
flexible infrastrucutre for continous multi-purpose data collection and analysis (P3)
cloud computing (M6)
additional database server (M5)
machine learning
ACA with R/Python (M4)
dedicated ACA programs (M3)
dictionaries/frequencies
methods and infrastructure
scaling up the method
String opertions in Excel/Stata/... (M2)
manual annotation
manual (M1)
scaling up the data set

Figure 1. Dimensions of scaling up content analysis.

In contrast, in a study that used simple SPSS string operations (M2) to count the occurrence of actors, but also words indicating concepts like "conflict" or "negativity", Vliegenthart, Boomgaarden, and Boumans (2011) were able to automatically code more than 400,000 articles without needing any human annotator, allowing them to study the prevalence of personalization, presidentialization, conflict, and negativity over time.

An example for ACAs using dedicated programs (M3) could be a large agenda-setting study by Zoizner, Sheafer, and Walgrave (2017), involving almost half a million news articles and speeches, and allowing them to answer the question how media coverage sets the agenda of politicians. For this, they used a proprietary program called Lexicoder. A limitation of such approaches is their inflexibility: Even though the list of indicators for specific concepts supplied by the program might have worked very well in their case, it can be problematic to adapt the methodology. For instance, if one wanted to use only slightly more complicated rules (e.g., X must be mentioned, but Y not), then it becomes hard or impossible to adapt such a program—but such an adjustment would be trivial if a language like R or Python is used. This is illustrated in a study by Kroon (2017): In a very similar fashion, she was interested in measuring frames in communication about older employees. However, she did not use fixed lists of words in a standalone program, but implemented the same functionality using regular expressions and AND-, OR-, as well as NOT-conditions.

Even though dedicated programs for ACA exist, Jacobi et al. (2016), who analyzed more than 50,000 news articles, note: "The easiest way to get started with LDA [a popular ACA technique; DT & JJ] is through the open-source statistical package R. Although specialized software for topic models is available […], an advantage of using R is that it is a statistical package that many social scientists already use for other analyses." (p. 95). Indeed, using an environment like R or

Python (M4) becomes increasingly popular among communication scientists. Another example might be a topic model of 77 million tweets processed by Guo, Vargo, Pan, Ding, and Ishwar (2016).

Many studies take scalability a step further an use a dedicated database in combination with R (e.g., Driscoll & Walker, 2014; Kleinnijenhuis, Schultz, & Oegema, 2015) or Python (e.g., Jonkman et al., 2016; Strycharz et al., 2018), which frequently is not run locally, but on a dedicated server (M5). These can be dedicated, physical servers, but also virtual machines on a cloud computing platform. The studies of Kleinnijenhuis et al. (2015) and Strycharz et al. (2018) both dealt with similar research questions: They were interested in the relationship between news and stock exchange rates. In such cases, the availability of a large-scale database, in which complicated searches can be done, and in which subsequent analyses can be executed in languages such as R (as in Kleinnijenhuis et al., 2015) or Python (as in Strycharz et al., 2018), pays off. A final example would be a study by Lansdall-Welfare, Sudhahar, Thompson, Lewis, and Cristianini (2017), who used distributed computing (M6) to analyze 150 years of British periodicals, which amounts to millions of articles, using Hadoop.

Moving from projects of a limited scope (P1) to larger projects (P3) also implies that different methods and different technical infrastructures come into play. As we can see, scaling up the size of the dataset (depicted on the lower x-axis) is related to scaling up the method (M1–M6, as denoted on the y-axis). The manually collected and annotated dataset (M1) is clearly not scalable. While dictionary and frequency-based methods can be used already with simple software (M2), the use of more advanced methods like machine learning requires more skills (usually M4) and shows its power mostly in larger datasets. While a dedicated content analysis software (M3) might already have some machine learning functions integrated, their capabilities are usually not scalable enough, for instance due to their inflexibility or limitations on the amount of documents they can process.

In other words: For a limited study (P1), a manual method (M1) may still be feasible. Once the scope of the projects grows, we move from the non-scalable manual approach (M1), via a slightly better scalable approach using common software packages (M2 and M3) and programming languages (M4) to the use of dedicated database servers (M5) and possibly cloud computing (M6).[7]

Proposing a scheme like the one in Figure 1 inherently loses nuance (e.g., supervised machine learning might require some manual annotation); however, it can provide a rough guide on what to take into account when planning to scale-up content analysis. Let us outline how a scaled-up content analysis that could be placed in the upper-right corner of Figure 1 might look like.

As an example, Figure 2 shows the general design for a system that can be used to conduct content analyses like the examples discussed above. It comprises of three steps: (1) retrieving, structuring, and storing the data; (2) cleaning the data; and (3) analyzing the data.

Structuring and storing the data

The researcher first has to decide on a format in which to store the data. To meet the criteria of scalability, independence, and adaptability, we choose an open format that allows us to store all kind of data—for example tweets, news items, and Facebook comments. We decide to use a JSON-based approach: an open standard to store key-value pairs in a human-readable way. It is flexible, supports nested data structures, and allows us to store full texts (in fact, any kind of data) in a database, along with other fields that provide some meta-information (e.g., date of

[7] We do not discuss specific frameworks like Hadoop in detail, because they are for even large-scale content analysis not (yet) neccessary (for an exception, see Lansdall-Welfare et al., 2017). This may change, however, once it becomes more common to analyze non-textual data. For a comparison between MongoDB and Hadoop for social media research, we refer to Murthy and Bowman (2014).

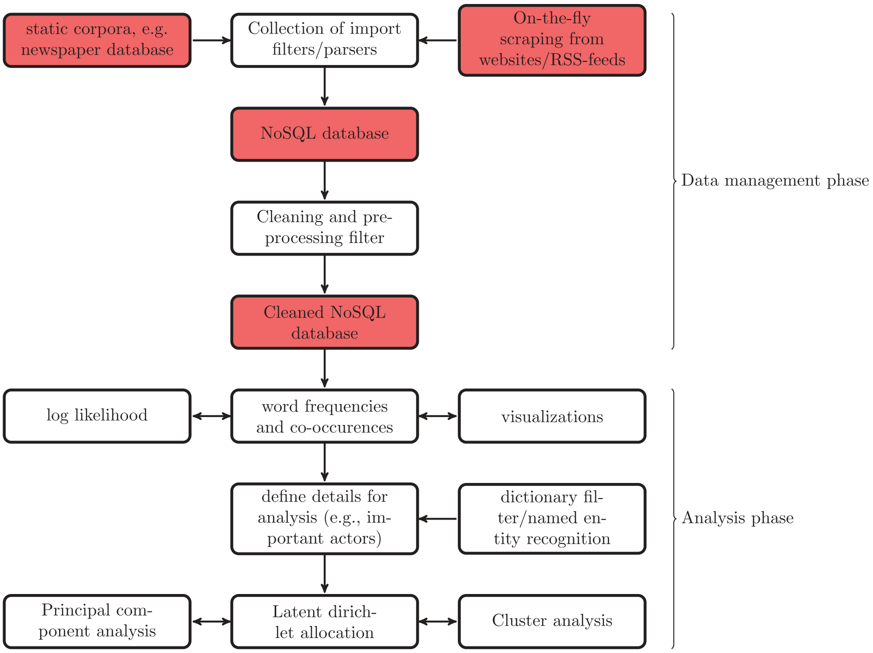
static corpora, e.g. newspaper database
Collection of import filters/parsers
On-the-fly scraping from websites/RSS-feeds
NoSQL database
Cleaning and pre-processing filter
Cleaned NoSQL database
Data management phase
log likelihood
word frequencies and co-occurences
visualizations
define details for analysis (e.g., important actors)
dictionary filter/named entity recognition
Analysis phase
Principal component analysis
Latent dirichlet allocation
Cluster analysis

Figure 2. An example of different phases in a scaled-up automated content analysis.

publication, source, length, language, etc.). We use an open-source database (like MongoDB or ElasticSearch) as backend to comply with our criteria (see Günther, Trilling, and Van de Velde (2018) for a discussion on the use of these databases in the social sciences).

Cleaning the data

In content analysis, it is crucial to avoid missing mentions of the core concepts regarding the topic of study. We address this issue in two ways. First, we use a preprocessing step in which possible variations of core concepts (e.g., Wal-Mart or Wal Mart) are replaced by a uniform code (e.g., Walmart). To do so, we propose using a priori listings of case-specific actors and concepts (see, e.g., Jonkman et al. (2016) for an implementation of this). These can be words, but also patterns. Also, things like part-of-speech tagging (POS) or named entity recognition (NER) can be performed at this stage. Because the results are fed back into the database, even operations that take a long time to run (maybe prohibitively long on an individual computer) can be used more efficiently, as it only has to be executed once. This is an important consideration for developing a truly scalable system, that at the same time avoids doing the same tasks over and over again. Because of its widespread use within the community, a good choice could be to implement these functions using Python (e.g., Freelon, 2014a; Günther & Quandt, 2016).

Analyzing the data

Communication researchers often use methods that are based on word frequencies—e.g., assessment of Log-Likelihood scores; Principal Component Analyses (PCA); co-occurrence or semantic networks, and cluster analyses (see, e.g., Van der Meer, 2016). Recently, a new tool has extended this collection: topic models like Latent Dirichlet Allocation (LDA) (for examples of LDA in communication research, see Günther & Quandt, 2016; Jacobi et al., 2016; Strycharz et al., 2018; Tsur et al.,

2015) and its extensions author-topic models and structured topic models. All these methods have their owns merits, and they are often used together to reach a deeper understanding. In particular, all of the analysis functions can be conducted using popluar open-source modules like the Python packages NLTK (Bird et al., 2009), scikit-learn (Pedregosa et al., 2011), and gensim (Řehůřek & Sojka, 2010), which makes the complete workflow transparent and reproducible. But while the analysis of texts if the prevalent use case in today's communication science, there is no inherent reason not to also use such methods to analyze pictures or other type of data, which could as well be stored within the system.

As this example shows, using only open-source components and components that as far as possible are suitable for later scaling up and that are integrated into a consistent workflow, we can build a system that conforms with the criteria we developed and falls in the right-upper corner of the scalability-scheme we presented in Figure 1.

A note on feasibility

Some might object that we sketched an "ideal world scenario" that proposes standards and guidelines that cannot be adhered to by colleagues and teams with limited means. However, we argue that the hurdles are surmountable.

Regarding our first point, scalability and the ability to run the infrastructure on a server, the resources needed are limited. Many universities or regional or national scientific platforms offer server infrastructure for free, and the event of commercial cloud computing platforms like Amazon Web Services allow renting the necessary resources for a couple of hundreds of euros per year—a negligible cost compared to the costs for human annotators in manual content analyses. What usually is necessary, though, is some knowledge of the Linux operating system—a skill that is not exactly rocket science.

Our second point, that all components should be free open source software (FOSS), obviously does not involve any financial investment. Regarding the third point, adaptability, the main investment lies in actually *doing* all these adoptions. For instance, looking at the steps outlined in Figure 2, probably one of the most time-consuming steps is the adoption of the import filters and parsers, as these need to be tailored to the specific types of data at hand. The necessary time for this, though, can be massively reduced by providing standard templates that people with limited training can adapt. While it is difficult to give an accurate estimate of the level of training that is needed, an indication might be provided that we regularly hire graduate students, which had no prior programming knowledge but followed an eight-week course on automated content analysis, to work on and enhance a framework like the one we presented.

Regarding the fourth point, the availability of multiple interfaces, is probably the most demanding one, while at the same time, one of the most important ones: fter all, ease-of-use is crucial to encourage the adoption of the system. While there are systems that feature easy-to-use web interfaces that allow untrained users to interact with the system (e.g., Borra & Rieder, 2014; Niekler & Wiedemann, 2015; Niekler et al., 2014; Van Atteveldt, 2008), developing such an interface takes a lot of time and asks for specific skills. What can help is a step-by-step approach, in which first ready-to-use components are used (like, e.g., the Kibana dashboard for the ElasticSearch database) as a temporal solution, before developing a custom web interface that unlocks the full potential also for users with less technical knowledge.

To sumamrize, it is probably safe to say that setting up a basic infrastructure that conforms to our guidelines—even if its done for the first time—can be done in a few days with limited effort and financial means, provided that one makes use of the available building blocks and systems out there (e.g., Van Atteveldt, 2008). Getting everything "right" and optimally adapted to the own needs and purposes, however, might take some months, and in particular hiring some research assistants.

Conclusion

In this article, we set out to present a framework for scaling up content analysis. In particular, we proposed four criteria: scalability, open source, adaptability, and multiple interfaces.

We demonstrated that these criteria can be fulfilled by setting up a framework relying on a NoSQL database (like MongoDB or ElasticSearch) as backend and a set of tools around it to effectively implement all functions needed for data collection, structuring, preprocessing, and analysis. Because of the wide availability of packages for text analysis and machine learning, implementing such a system in the Python programming language could be a good choice. However, we would like to stress that our argument is language-agnostic: others might choose for other backends and languages.

In doing so, we reflected on the state of the art and aimed at providing a road map for the field. While there is an increasing interest in automated content analysis techniques, many current approaches lack scalability. While standalone packages offer ease-of-use, this limitation leads to automated content analysis falling short of its possibilities. Given the growing role of large-scale data analysis in social science research (see for an extensive discussion Kitchin, 2014a), it is of crucial importance to have the necessary infrastructure and one's disposal. And here, communication scientists should be at the forefront: as Big Data analysis requires not only technical skills, but also thorough knowledge of the field (Kitchin, 2014a, p. 162), the design of a framework for content analysis is not merely an engineering problem that could be outsourced or bought in.

Numerous subfields in communication science can benefit from using a framework as we described in Figure 2. For instance, continuously accumulating texts from different sources, like news media, social media, or press releases, would allow to conduct analyses that are difficult or impossible to conduct with smaller samples, such as answering questions about framing or agenda-setting effects on a fine-grained level. Most actors are actually *not* in the news on most days, so studying them requires enormous amounts of data. Being able to use ACA techniques to identify topics in such a corpus and to study their development over time allows us to answer exciting questions around news hypes and dynamics of news dissemination.

The relevance of having a flexible and scalable framework is even more apparent when analyzing user-generated comment like product reviews, comments on articles, and so on. This kind of data, which is increasingly of interest to researchers from various fields, has numerous properties which call for a framework as we described. In particular, their nested structure (reviews belonging to a product, comments belonging to an article), their highly diverse structure, and their ever-growing volume make it infeasible to use off-the-shelf software. Moreover, to study personalized content (i.e., different users getting a different version of an article, an advertisement, and so on) it is necessary to have a technological framework that allows storing the different versions and comparing their similarities and differences.

To do all this, however, communication scientists have to break with several habits. First of all, they must shift the project management focus toward more long-term data collection and re-usability of data and infrastructure across projects. In manual content analysis, the cost associated with data collection and analysis is roughly proportional to the amount of data. Apart from relatively low startup costs for things like coder training, it essentially is twice as expensive to code twice as many articles. In contrast to this, automated content analysis can have a rather high startup cost, but doubling the amount of data does not cost much. This is why it should be avoided to invest in project-specific solutions, and why an extensible, flexible, and open framework geared towards re-usability should be preferred. Second, communication scientists have to let go of the idea of analyzing a dataset represented by *one* file. Instead, as illustrated in our figures, they should switch to using databases, which can be updated with new data, re-used for multiple projects, and also store intermediate results (like preprocessed text). Third, they should define the scope of their needs according to a scheme like the one we presented in Figure 1. While they might not end up in the upper right corner, which is the most heavily scaled-up approach, they should make an informed decision of how much scaling up is feasible and necessary for them.

Our notion of scaling up content analysis therefore should be seen as a guiding principle and aid for researchers that are in the adopting automated content analysis techniques. The principle of scaling up implies that already at the design phase one should think about how to re-use data and analysis, thereby focusing on the scalability of the approach. Scalability does not only refer to storage size of computational power, but also to things like making it scriptable, and to the avoidance of manual interventions. Manually copy-pasting something might be fine for one file, but what if you have hundreds of them?

For future work, the line of reasoning to behind our criteria for a framework for scaled-up content analysis can be extended. For instance, one can think of making sure that each module in such a framework (input filters, text-processing facilities, analyses) is implemented in a standardized way, to make sure that people with minimal programming knowledge can add or modify these elements (think of student assistants who followed a methods course). This would also make it easier to increase the number of modules to make it more useful for more colleagues to create synergy effects.

An additional way to extend an implemention as discribed in Figure 2 is to follow the example of frameworks which offer the possibility for human *coders* or *annotators* (in communication science lingo and computer science lingo, respectively) to manually code or annotate content (e.g., Niekler & Wiedemann, 2015; Van Atteveldt, 2008). This can be either useful in itself (if the automated part of the analysis is only meant to reduce the number of relevant articles), or it can be used as input to train a machine learning algorithm, which then classifies the rest of the material automatically.

Strycharz et al. (2018) used such a function: They used a regular expression-based search query to identify potentially relevant articles in their database backend, but the retrieved articles where then presented to human annotators who could tag the articles as either relevant or not. While this approach, obviously, does not scale well, it nevertheless could be interesting as an additional component in a scalable framework: Because the annotators' tags where stored in the database itself, one could try to train a supervised machine learning algorithm on it to improve future search results, which, in fact, would add to the system's scalability.

Seeing the increased use of automated content analysis in communication science, we hope to see more implementations of systems that allow scaling up content analysis—but even more, we hope to spark a discussion on best practices for scaling up content analysis. After all, the journey has just begun.

Funding

This work was carried out on the Dutch national e-infrastructure with the support of SURF Foundation.

ORCID

Damian Trilling http://orcid.org/0000-0002-2586-0352

References

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

Borra, E., & Rieder, B. (2014). Programmed method: Developing a toolset for capturing and analyzing tweets. *Aslib Journal of Information Management*, *66*(3), 262–278. doi:10.1108/AJIM-09-2013-0094

Boumans, J. W. (2016). *Outsourcing the news? An empirical assessment of the role of sources and news agencies in the contemporary news landscape* (Ph.D. dissertation, University of Amsterdam). Retrieved from http://hdl.handle.net/11245/1.532941

Boumans, J. W., & Trilling, D. (2016). Taking stock of the toolkit: An overview of relevant autmated content analysis approaches and techniques for digital journalism scholars. *Digital Journalism*, *4*(1), 8–23. doi:10.1080/21670811.2015.1096598

Broussard, M. (2016). Big data in practice. *Digital Journalism*, *4*, 266–279. doi:10.1080/21670811.2015.1074863

Burggraaff, C., & Trilling, D. (2017). Through a different gate: An automated content analysis of how online news and print news differ. *Journalism*. doi:10.1177/1464884917716699

Burscher, B., Odijk, D., Vliegenthart, R., de Rijke, M., & de Vreese, C. H. (2014). Teaching the computer to code frames in news: Comparing two supervised machine learning approaches to frame analysis. *Communication Methods and Measures*, *8*(3), 190–206. doi:10.1080/19312458.2014.937527

Burscher, B., Vliegenthart, R., & De Vreese, C. H. (2015). Using supervised machine learning to code policy issues: Can classifiers generalize across contexts? *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 122–131. doi:10.1177/0002716215569441

Busch, L. (2014). A dozen ways to get lost in translation: Inherent challenges in large-scale data sets. *International Journal of Communication*, *8*, 1727–1744.

Cioffi-Revilla, C. (2014). *Introduction to computational social science: Principles and applications*. London, UK: Springer.

De Rooij, O., Vishneuski, A., & De Rijke, M. (2012). xTAS: Text analysis in a timely manner. *Proceedings of the 12th Dutch-Belgian Information Retrieval Workshop (DIR 2012*), 89–90.

Driscoll, K., & Walker, S. (2014). Working within a black box: Transparency in the collection and production of big twitter data. *International Journal of Communication*, *8*, 1745–1764.

Freelon, D. (2014a). On the cutting edge of big data: Digital politics research in the social computing literature. In S. Coleman, & D. Freelon (Eds.), *Handbook of digital politics*. Northampton, MA: Edward Elgar.

Freelon, D. (2014b). On the interpretation of digital trace data in communication and social computing research. *Journal of Broadcasting & Electronic Media*, *58*(1), 59–75. doi:10.1080/08838151.2013.875018

Gerbner, G. (1983). The importance of being critical—In one's own fashion. *Journal of Communication*, *33*(3), 355–362. doi:10.1111/j.1460-2466.1983.tb02435.x

Gonzalez-Bailon, S., & Paltoglou, G. (2015). Signals of public opinion in online communication: A comparison of methods and data sources. *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 95–107. doi:10.1177/0002716215569192

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., & Quandt, T. (2016). Word counts and topic models. *Digital Journalism*, *4*(1), 75–88. doi:10.1080/21670811.2015.1093270

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

Guo, L., Vargo, C. J., Pan, Z., Ding, W., & Ishwar, P. (2016). Big social data analytics in journalism and mass communication: Comparing dictionary-based text analysis and unsupervised topic modeling. *Journalism & Mass Communication Quarterly*, *93*, 332–359. doi:10.1177/1077699016639231

Heiberger, R. H., & Riebling, J. R. (2016). Installing computational social science: Facing the challenges of new information and communication technologies in social science. *Methodological Innovations*, *9*, 1–11. doi:10.1177/2059799115622763

Hellsten, I., Dawson, J., & Leydesdorff, L. (2010). Implicit media frames: Automated analysis of public debate on artificial sweeteners. *Public Understanding of Science*, *19*(5), 590–608. doi:10.1177/0963662509343136

Hindman, M. (2015). Building better models: Prediction, replication, and machine learning in the social sciences. *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 48–62. doi:10.1177/0002716215570279

Jacobi, C., Van Atteveldt, W., & Welbers, K. (2016). Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, *4*(1), 89–106. doi:10.1080/21670811.2015.1093271

Jonkman, J. G. F., Trilling, D., Verhoeven, P., & Vliegenthart, R. (2016). More or less diverse: An assessment of the effect of attention to media salient company types on media agenda diversity in Dutch newspaper coverage between 2007 and 2013. *Journalism, Online First*. doi:10.1177/1464884916680371

Jonkman, J. G. F., & Verhoeven, P. (2013). From risk to safety: Implicit frames of third-party airport risk in Dutch quality newspapers between 1992 and 2009. *Safety Science*, *58*, 1–10. doi:10.1016/j.ssci.2013.03.012

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

Kitchin, R. (2014b). *The data revolution: Big data, open data, data infrastructures and their consequences*. London, UK: Sage.

Kleinnijenhuis, J., Schultz, F., & Oegema, D. (2015). Frame complexity and the financial crisis: A comparison of the United States, the United Kingdom, and Germany in the period 2007–2012. *Journal of Communication*, *65*(1), 1–23. doi:10.1111/jcom.12141

Krippendorff, K. (2004). *Content analysis: An introduction to its methodology* (2nd ed.). Thousand Oaks, CA: Sage.

Kroon, A. C. (2017). *Images of older workers: Content, causes, and consequences* (PhD dissertation, University of Amsterdam). Retrieved from http://hdl.handle.net/11245.1/0980ab70-3251-498a-b5a3-9bc288023062

Lacy, S., Watson, B. R., Riffe, D., & Lovejoy, J. (2015). Issues and best practices in content analysis. *Journalism & Mass Communication Quarterly*, 1–21. doi:10.1177/1077699015607338

Lansdall-Welfare, T., Sudhahar, S., Thompson, J., Lewis, J., & Cristianini, N. (2017). Content analysis of 150 years of British periodicals. *Proceedings of the National Academy of Sciences*, *114*(4), E457–E465. doi:10.1073/pnas.1606380114

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

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

Leydesdorff, L., & Welbers, K. (2011). The semantic mapping of words and co-words in contexts. *Journal of Infometrics*, *5*(3), 469–475. doi:10.1016/j.joi.2011.01.008

Leydesdorff, L., & Zaal, R. (1988). Co-words and citations relations between document sets and environments. In *Informetrics* (pp. 105–119). Elsevier.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) system demonstrations* (pp. 55–60). Retrieved from http://www.aclweb.org/anthology/P/P14/P14-5010

Mayer-Schönberger, V., & Cukier, K. (2013). *Big Data: A revolution that will transform how we live, work, and think*. Boston, MA: Houghton Mifflin Harcourt.

Murthy, D., & Bowman, S. A. (2014). Big Data solutions on a small scale: Evaluating accessible high-performance computing for social research. *Big Data & Society*, *1*(2), 1–12. doi:10.1177/2053951714559105

Niekler, A., & Wiedemann, G. (2015). Semi-automatic content analysis for the identification of neo-liberal justifications in large newspaper corpora. In *GESIS Computational Social Science Winter Symposium*. doi: 10.13140/RG.2.1.2283.1128

Niekler, A., Wiedemann, G., & Heyer, G. (2014). Leipzig Corpus Miner – A text mining infrastructure for qualitative data analysis. In *Terminology and knowledge engineering 2014 (TKE 2014)*. Berlin.

Parks, M. R. (2014). Big Data in communication research: Its contents and discontents. *Journal of Communication*, *64*(2), 355–360. doi:10.1111/jcom.12090

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.

Řehůřek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45–50). Valletta, Malta: ELRA. http://is.muni.cz/publication/884893/en

Russell Neuman, W., Guggenheim, L., Mo Jang, S., & Bae, S. Y. (2014). The dynamics of public attention: Agenda-setting theory meets Big Data. *Journal of Communication*, *64*(2), 193–214. doi:10.1111/jcom.12088

Scharkow, M. (2011). Thematic content analysis using supervised machine learning: An empirical evaluation using German online news. *Quality & Quantity*, *47*(2), 761–773. doi:10.1007/s11135-011-9545-7

Schnell, R., Bachteler, T., & Reiher, J. (2005). MTB: Ein Record-Linkage-Programm für die empirische Sozialforschung. *ZA-Information*, *56*, 93–103.

Schuck, A. R., Xezonakis, G., Elenbaas, M., Banducci, S. A., & De Vreese, C. H. (2011). Party contestation and Europe on the news agenda: The 2009 European parliamentary elections. *Electoral Studies*, *30*(1), 41–52. doi:10.1016/j.electstud.2010.09.021

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

Sjøvaag, H., & Stavelin, E. (2012). Web media and the quantitative content analysis: Methodological challenges in measuring online news content. *Convergence: the International Journal of Research into New Media Technologies*, *18*(2), 215–229. doi:10.1177/1354856511429641

Social Media and Political Participation Lab at New York University. (2016). *smapPy*. Retrieved from https://github.com/SMAPPNYU/smappPy

Strycharz, J., Strauss, N., & Trilling, D. (2018). The role of media coverage in explaining stock market fluctuations: Insights for strategic financial communication. *International Journal of Strategic Communication*, *12*(1), 67–85. doi:10.1080/1553118X.2017.1378220

Trilling, D., Tolochko, P., & Burscher, B. (2017). From newsworthiness to shareworthiness. *Journalism & Mass Communication Quarterly*, *94*(1), 38–60. doi:10.1177/1077699016654682

Tsur, O., Calacci, D., & Lazer, D. (2015). A frame of mind: Using statistical models for detection of framing and agenda setting campaigns. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing* (pp. 1629–1638). ACL.

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

Van der Meer, G. L. A. (2016). *Communication in times of crisis: The interplay between the organization, news media, and the public* (PhD thesis, University of Amsterdam). Retrieved from http://hdl.handle.net/11245/1.532222

Van der Meer, G. L. A., Verhoeven, P., Beentjes, H., & Vliegenthart, R. (2014). When frames align: The interplay between PR, news media, and the public in times of crisis. *Public Relations Review*, *40*(5), 751–761. doi:10.1016/j.pubrev.2014.07.008

Vliegenthart, R., Boomgaarden, H. G., & Boumans, J. (2011). Changes in political news coverage: Personalisation, conflict and negativity in British and Dutch newspapers. In K. Brants, & K. Voltmer (Eds.), *Political communication in postmodern democracy: Challenging the primacy of politics* (pp. 92–110). London, UK: Palgrave Macmillan.

Welbers, K., van Atteveldt, W., Kleinnijenhuis, J., & Ruigrok, N. (2016). A gatekeeper among gatekeepers. *Journalism Studies, Online First*. doi:10.1080/1461670X.2016.1190663

Zamith, R., & Lewis, S. C. (2015). Content analysis and the algorithmic coder: What computational social science means for traditional modes of media analysis. *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 307–318. doi:10.1177/0002716215570576

Zoizner, A., Sheafer, T., & Walgrave, S. (October 2017). How politicians' attitudes and goals moderate political agenda setting by the media. *The International Journal of Press/Politics*, *22*(4), 431–449. doi:10.1177/1940161217723149