Using the HTRC Data Capsule Model to Promote Reuse and Evolution of Experimental Analysis of Digital Library Data: A Case Study of Topic Modeling

David Bainbridge
University of Waikato
Hamilton, New Zealand
davidb@waikato.ac.nz

David M. Nichols
University of Waikato
Hamilton, New Zealand
david.nichols@waikato.ac.nz

Annika Hinze
University of Waikato
Hamilton, New Zealand
hinze@waikato.ac.nz

J. Stephen Downie
University of Illinois
Urbana-Champaign, USA
jdownie@illinois.edu

ABSTRACT

We report on a case-study to independently reproduce the work given in a publicly available blog on how to develop a topic model sourced from a collection of texts, where both the data set and source code used are readily available. More specifically, we detail the steps necessary—and the challenges that had to be overcome—to replicate the work using the HathiTrust Research Center's virtual machine Data Capsule platform. From this we make recommendations for authors to follow, based on the lessons learned. We also show that the Data Capsule model can be put to work in a way that is of benefit to those interested in supporting computational reproducibility within their organizations.

CCS CONCEPTS

• **General and reference → Experimentation**; • **Applied computing → Digital libraries and archives**.

KEYWORDS

Experimental Reproducibility, Virtual Machine, Digital Libraries

ACM Reference Format:
David Bainbridge, David M. Nichols, Annika Hinze, and J. Stephen Downie. 2019. Using the HTRC Data Capsule Model to Promote Reuse and Evolution of Experimental Analysis of Digital Library Data: A Case Study of Topic Modeling . In *Proceedings of ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'19).* ACM, New York, NY, USA. 463-4. https://doi.org/10.1109/JCDL.2019.00124

1 INTRODUCTION

The scientific tenet of reproducibility has its own particular set of of challenges to be faced in the domain of computational science and related areas, such as digital humanities [2]. Virtualization techniques, such as Virtual Machines and Containerization, help by reducing the likelihood of failure caused by installation issues or effects from a user's environment, but as Dumas *et al.* point out [1], this really only establishes a baseline. The chances of reproducibility success, they point out, are greatly enhanced through providing a walk-through guide of the steps to run and what to expect as a result at each stage.

In this paper we narrow attention to the more specific issue of replication of data analysis experiments using corpora from digital libraries and archives. Taking the HathiTrust Research Center's (HTRC) Data Capsule [3], as a baseline virtual machine, we report on our experiences seeking to follow one such example of step-by-step instructions, to produce a topic model based on a publicly available texts. From this we make recommendations based on the lessons learned, and identify particular benefits that result from conducting the experimentation inside a Data Capsule.

2 REPLICATING TOPIC MODELING WITH A DATA CAPSULE

In *Creating a Topic Browser of HathiTrust Data*,[1] Goodwin (an English professor and practicing digital humanities scholar) provides a "how-to" article, describing the steps he went through to develop a web-based topic browser of a set of selected novels from the HathiTrust DL. The article is written in a follow-along style: the starting dataset used is linked to, as is the R programming language open source Latent Dirichlet Allocation (LDA) package that the approach draws upon, and it also provides the additional lines of code written to produce the topic browser. The description of the work is agnostic as to the operating system used, although the way files are specified suggests Goodwin used a Unix-based operating system. These characteristics made it a suitable candidate for our replication study using an Ubuntu-based HTRC Data Capsule.

In describing below the problems we faced, we in no way want this to be taken as a criticism of Goodwin's excellent work. To the contrary, we very much appreciated the comprehensive notes provided. As pragmatists we knew it likely that we would encounter issues: what we were interested in, however, was what forms they would take, and the strategies we could develop to overcome them.

[1] *https://jgoodwin.net/blog/creating-hathitrust-topic-browser/*

JCDL'19, June 2019, Urbana-Champaign, Illinois, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1109/JCDL.2019.00124

We categorize the five principal difficulties encountered as follows:

(1) Link-rot.
(2) Installation clarity.
(3) Incorrect file/directories specified.
(4) Version of programming language and packages used.
(5) Issues running commands more than once.

There was only one example of link-rot that we encountered, and that was to the general web area where the dataset used came from, resulting in a blank gray page. Fortunately, the link to the precise dataset used in the article still worked, and so was not a critical issue.

The principal programming language used was R, and this was clearly stated, as was the use of Perl for some file manipulation operations. What was not stated was that because installation of some of the R packages themselves trigger the compilation of Java and C compiled code, then these too needed to be installed. Further, this compilation sequence relied on certain libraries being present. As the Data Capsule environment provides administration rights for the user, these unexpected requirements could be addressed in a straightforward manner.

The article helpfully included the exact lines of code to run. However, on more than one occasion, the name of a file or directory used in the code-snippets was inconsistently specified: for example `sample-00-22-tsv` is used in one place for the output directory of tab-separated value files, but `20-22-tsv` is used in another. These issues were not too hard to resolve.

The most time-consuming problem encountered concerned the LDA topic modeling R package used, `dfrtopics`. Ultimately the problem was traced to be a versioning issue. In the instructions given, this package was retrieved through a *github* repository link that resolved to the latest version of that code; however, updates in the *github* repository meant that the code checked out was now incompatible with how Goodwin invoked it. The date of the blog article was cross-checked with the versioning history of the `dfrtopics` *github* repository, and the R statement to install the package changed to check-out a contemporaneous version.

The final main issue encountered—that of re-running commands—was itself an artifact caused by the problems that were encountered in trying to follow the full set of instructions. For example, we encountered a situation where executing a step reported an error if the directory it wanted to write to already existed. A single Linux command was sufficient to rectify this, but it served to highlight the potential for such issues occurring in the other stages, but in ways that were not explicitly reported.

With all the issues resolved, we deleted all the generated output files and were able to take a clean pass through the instructions and generate our own topic map. Given that Goodwin's article links to a live version of the topic browser he built, we could see that the topic browser we had produced was not the same. For the sake of transparency, we do note that the blog article does not claim that by following the instructions given you will achieve identical results. That said, there is sufficient detail in the blog to understand the main reasons for the differences.

In the blog article, the number of topic clusters to produce is set to be 100. In Goodwin's live example he has 125 topics. Another source of difference could be the number of iterations used to train the model: the blog article specifies this to be 200. It was not possible to determine how many iterations had been used in the live example.

A final factor that led to notable differences in the topics generated was a result of the stopword list used. The actual stopword list used by Goodwin is described but not explicitly provided. The article mentions a stopword list contained in `dfrtopics`, which we located and used, but it is not clear if this is the same one as used by Goodwin.

3 LESSONS LEARNED AND CONCLUSION

Based on the lessons learned, we recommend authors give:

- Careful consideration to the URLs used when publishing links to datasets, and even consider using the Internet Archive's Wayback Machine to provide better link stability.
- A how-to guide a "test-drive" by a second researcher (akin to proof-reading). This would also help address installation clarity.
- An explicit list of all the programming languages and packages used, along with their version numbers.

Following these recommendations, to augment the blog article by Goodwin, we have formed our own *github* repository that explains how to work through the article using an HTRC Data Capsule: *https://github.com/htrc/JGoodwin-Topic-Browser-in-a-Data-Capsule/*

We conclude by highlighting two key benefits of the HTRC Data Capsule model deployment over working directly with standalone VMs:

Readiness to run. Even as experienced VirtualBox users it still took us 20 minutes to set up a VM to the point where we could follow the blog article, compared to filling out a form at the HTRC Analytics site and clicking on a button to create the Data Capsule, which had us at that point in under 3 minutes.

Network integrity. There are security issues that an institute must work through if a researcher is going to operate VM software directly on their own computer. In the infrastructure developed by HTRC to run Data Capsules, these networking concerns have already been resolved.

Without doubt, the use of walk-through guides in combination with VMs greatly aids computational reproducibility. Our particular use of the HTRC Data Capsule illustrates the general benefits of the Data Capsule model that other institutions should consider deploying if they are interested in supporting computational reproducibility.

REFERENCES

[1] Guillaume Dumas, Yang-Min Kim, and Jean-Baptiste Poline. 2018. Experimenting with reproducibility: a case study of robustness in bioinformatics. *GigaScience* 7, 7 (06 2018), 1–8. https://doi.org/10.1093/gigascience/giy077 arXiv:http://oup.prod.sis.lan/gigascience/article-pdf/7/7/giy077/25178539/giy077.pdf

[2] Roger D. Peng. 2011. Reproducible Research in Computational Science. *Science* 334, 6060 (2011), 1226–1227. https://doi.org/10.1126/science.1213847 arXiv:http://science.sciencemag.org/content/334/6060/1226.full.pdf

[3] Jiaan Zeng, Guangchen Ruan, Alexander Crowell, Atul Prakash, and Beth Plale. 2014. Cloud Computing Data Capsules for Non-consumptive use of Texts. In *Proceedings of the 5th ACM Workshop on Scientific Cloud Computing (ScienceCloud '14)*. ACM, New York, NY, USA, 9–16. https://doi.org/10.1145/2608029.2608031