Reproducibility in Computational Linguistics: Is Source Code Enough?

Mohammad Arvan, Luís Pina, and **Natalie Parde**
Department of Computer Science
University of Illinois Chicago
{marvan3,luispina,parde}@uic.edu

Abstract

The availability of source code has been put forward as one of the most critical factors for improving the reproducibility of scientific research. This work studies trends in source code availability at major computational linguistics conferences; namely, ACL, EMNLP, LREC, NAACL, and COLING. We observe positive trends, especially in conferences that actively promote reproducibility. We follow this by conducting a reproducibility study of eight papers published in EMNLP 2021, finding that source code releases leave much to be desired. Moving forward, we suggest all conferences require self-contained artifacts and provide a venue to evaluate such artifacts at the time of publication. Authors can include small-scale experiments and explicit scripts to generate each result to improve the reproducibility of their work.

1 Introduction

Modern natural language processing and computational linguistics research progresses at a rapid pace, making it expedient to build upon the groundwork of earlier research rather than reinventing solutions from scratch. This is complicated by the field's reliance in recent years on deep learning models that are difficult to interpret and sensitive to small changes in architecture and environment. These distinct characteristics hinder their reproducibility, as do the substantial computing resources often required to replicate them. Thus, many strong NLP models fall short on two crucial parameters: accessibility and reproducibility.

Although deep learning models today allow for effective processing in highly complex search spaces and in most cases outperform solutions from the past, researchers must consider the risks and potential ethical implications associated with their use alongside the performance benefits. There have been numerous calls and attempts (Dodge et al., 2019; Rogers et al., 2021) to push the community into taking more responsibility for the reproducibility of their research. Several venues have introduced and adapted reproducibility checklists and standards (Pineau, 2019; Stojnic, 2022) into their submission process (Nature, 2022; NeurIPS, 2022; AAAI, 2022; ACM, 2022; ACL, 2022a; Deutsch et al., 2022), and others have organized reproducibility challenges (Sinha et al., 2021; Belz et al., 2021) to encourage the community to reproduce published research. Although this is undeniably a step forward, it is unclear how large of a step it has been. There is also some concern that asking reviewers to evaluate reproducibility burdens them with another time-consuming task that might extend beyond their expertise.

In this work, we analyze and report on the state of reproducibility in NLP. We investigate the extent to which content necessary for reproducing research is currently available, and study the influence of reproducibility checklists on this availability over the last seven years. We also conduct an eight-paper reproducibility case study to develop a deeper understanding of current strengths and weaknesses in research reproducibility in NLP. Our key contributions include the following:

- We scrape the ACL Anthology for data associated with all papers published at major venues in the last seven years. Then, we investigate the impact of the introduction of *reproducibility checklists* into the paper submission process, by analyzing trends in the source code availability of the accepted papers.
- We randomly select eight papers from the *2021 Conference on Empirical Methods in Natural Language Processing* (Moens et al., 2021a, EMNLP 2021) and attempt to reproduce their reported results. We find that despite the recent progress towards reproducibility, most released artifacts are of low quality. We make the artifacts from our own repro-

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 2350–2361
December 7-11, 2022 ©2022 Association for Computational Linguistics

ductions publicly available as self-contained Docker containers.

- We propose four recommendations to address major issues affecting reproducibility.

Our recommendations include incorporating small-scale experiments in papers to increase accessibility, creating well-documented scripts to reproduce reported results, publishing executable self-contained artifacts, and embedding artifact evaluation in the publication pipeline. We elaborate on our study process and findings in the remainder of this paper. Our hope is that this paper serves as a call to action for systemic improvements to research reproducibility in NLP.

2 Background

Numerous studies have assessed the reproducibility of scientific publications. This task often involves attempting to achieve results *close enough* to the ones reported in the paper with little to no reliance on the released software artifacts, if available. Raff (2019) attempts to quantify the reproducibility ratio of 255 papers published from 1984 to 2017. He selects different thresholds for a minimal acceptable error for algorithmic and empirical claims, ultimately reporting a 63% reproducibility ratio. In a similar study, Wieling et al. (2018) survey 395 papers presented at the ACL 2011 and 2016 conferences and identify whether links to data and code were provided. Then, they attempt to reproduce the results of ten papers using provided code and data. They ultimately find results close to those reported for six papers.

Olorisade et al. (2017) attempt to independently investigate the claims of six studies on text mining for citation screening. In the authors' words, 27% of machine learning papers lack the necessary information required for achieving reproducible results; hence, they introduce a checklist to help mitigate this issue. The challenge of dealing with missing information has also been brought up by Gundersen and Kjensmo (2018). Utilizing checklists and guiding authors towards better standards during the paper submission process has become a common practice amongst several venues (Nature, 2022; AAAI, 2022; ACM, 2022; ACL, 2022b; Pineau et al., 2021). Aside from guidelines, communities have organized reproducibility challenges (Sinha et al., 2021; Belz et al., 2021) that attempt to promote improved reproducibility across the field. While this increased attention towards openness and availability is a welcome change, the lack of consensus on terms and definitions has diminished progress.

There are a variety of definitions of and perspectives on reproducibility. Rougier et al. (2017) define *reproducing* as running the same software on the same input data and obtaining the same results. *Replicating* then is limited to running new software and achieving results judged as similar enough by an expert in the field. The Association for Computing Machinery (ACM) (ACM, 2022) considers the team and the experimental setup as contributing factors; furthermore, they add another term, *repeatability*, to the glossary of definitions. Whitaker (2017) and Schloss (2018) introduced two additional concepts known as *robustness* and *generalisability* to cover other missing dimensions.

These definitions attempt to cover an open-ended number of dimensions. Therefore, they often do not age well and become obsolete upon the arrival of a more comprehensive definition (Belz et al., 2022). Belz et al. (2022) suggest the community should adapt the definitions provided by the International Vocabulary of Metrology (JCGM, 2012, VIM). VIM defines reproducibility as a measurement precision under reproducibility conditions of measurement. These conditions must be known and recorded and include but are not limited to the source code, hyperparameters, dependencies, and runtime environment. As a result, this framework enables the use of precision as a measure of statistical variability to quantify how close results are to one another. Doing so will provide far more information than just binary assessments of whether research is reproducible or irreproducible.

3 Artifacts and Reproducibility

One of the ways to assure reproducibility of results in the broad field of computer science is through *research artifacts*. These are self-contained packages that contain everything needed to reproduce results reported in a paper, including: source code for research prototypes, scripts to build the source code, scripts to run the experiments reported in the paper, input data used in each experiment, experimental data obtained by the authors and reported in the paper, and scripts to process the experimental data and generate the tables and graphs in the paper. Furthermore, such artifacts are typically *self-executable*, meaning that they are packaged within a virtual machine (*e.g.,* VMWare or Virtual-Box) or within a container (*e.g.,* Docker) so that they run “out-of-the-box” on any machine. Thus, research artifacts are much more than simply releasing the source code for the prototype on a public website (*i.e.,* GitHub).

Conferences in other fields of computer science (*e.g.,* the 2021 ACM SIGPLAN International Conference on Object-Oriented Programming Systems, Languages, and Applications (Wadler and Drossopoulou, 2021, OOPSLA), the 2022 ACM SIGPLAN Conference on Programming Language Design and Implementation (Jhala and Dillig, 2022, PLDI), the 2022 European Conference on Object-Oriented Programming (Ali and Vitek, 2022, ECOOP), and the 2021 ACM International Conference on Architectural Support for Programming Languages and Operating Systems (Falsafi et al., 2022, ASPLOS)) allow authors to submit artifacts, typically after the paper is accepted. The quality of the submitted artifacts is then assessed by the *artifact evaluation committee (AEC)*, who run the artifacts following the authors’ instructions to judge whether the artifact reproduces all the results reported in the paper, and whether the artifact can be reused for future research on the same subject. At the recommendation of the AEC, published papers then get a set of badges on the first page. As a result, reproducing and building on research papers with an AEC badge should be straightforward: users should simply need to download the artifact and follow the instructions.

Conferences in the broad areas of NLP and machine learning have recently adopted *reproducibility checklists* (AAAI 2022 (AAAI, 2022), NeurIPS 2022 (NeurIPS, 2022)) which require authors to provide an appendix with the source code used in the experiments. Additionally, they recommend the inclusion of dependency specifications, training and evaluation code, pre-trained models, and a README file containing the information required to achieve the results in the paper. Unfortunately, as we show in the following sections, reproducing publications that follow these good practices is not easy. It requires a considerable amount of engineering involving fixing compilation errors, "guessing" configurations not documented, figuring out which code to execute for which experiment, manually processing experimental data, and dealing with obsolete tools or libraries.

Given the numerous challenges one faces to reproduce a research paper’s results, there has been a recent surge of venues solely focused on reproducibility (*e.g.,* the ML Reproducibility Challenge 2021 (Sinha et al., 2021) and ReproGen 2021 (Belz et al., 2021)), in which participants attempt to reproduce results of a selected published paper. Unlike the research artifacts described above, these reproducibility efforts do not lead to a self-contained and self-executable artifact. Self-contained and self-executable artifacts should be capable of reproducing the results in short order while avoiding vulnerability to “bit rotting” as time passes and widely available tools and libraries grow obsolete.

4 Definitions and Methods

We adapt the Metrology-based Reproducibility Assessment originally proposed by Belz et al. (2022) to achieve our reported results. In this framework, *reproducibility* is a measurement of precision and is directly connected to the conditions in which the measurement is being recorded. In the context of machine learning and natural language processing, these conditions include but are not limited to source code, trained models, evaluation methods, and datasets. Furthermore, *repeatability* is defined as a special case of reproducibility where the conditions across different measurements are the same. This formulation enables the use of common terms used in reporting precision. We report and focus on coefficient of variation (CV^*), defined as the unbiased sample standard deviation over the mean.

Ultimately, the gold standard for scientific artifacts is to have the highest level of reproducibility precision (or low variability). However, this standard is not practical. An alternative approach is to gradually push the community towards improving existing reproducibility standards. Existing reproducibility tracks are a great example of this effort. One of the primary objectives of the reproducibility movement is to increase availability of data and source code. Source code is an integral part of the reproducibility process since it allows other researchers to implement or execute conditions and details that may have been omitted in the publication itself. Here, we investigate trends in source code availability for scientific papers published in major conferences of the Association for Computational Linguistics (ACL) in recent years, using information from the ACL Anthology. Observing these trends is especially interesting since not all ACL conferences have introduced a reproducibility track in their submission process. We expected to observe a higher code availability ratio among venues that actively promote reproducibility.

Following this, we switch our focus and investigate whether the released source code is enough to reproduce the reported results. Empirical results have become harder to reproduce over time for known and unknown reasons (several of these reasons are discussed in Section 5.2). Although we quantify reproducibility using CV^*, we consider a reproducibility attempt successful if we are able to achieve any results. We understand that this process is not comprehensive—we cannot determine *absolute* reproducibility, since one may be able to spend days debugging a specific source code to ultimately achieve the reported results. However, absolute reproducibility and its time cost is not feasible at scale or for many researchers. In our case study, we clearly explain all steps taken by following instructions provided by the authors. We raise any questions or issues we face along the way, opening direct lines of communication with authors to overcome challenges and improve the reproducibility of the released source code as part of our process. We hypothesize that despite the increased availability of source code for papers at large NLP conferences, achieving reproducible results may still be an extremely challenging task.

For our case study, we randomly selected eight papers from the *2021 Conference on Empirical Methods in Natural Language Processing* (Moens et al., 2021a). We tried to reproduce their results using the code and instructions provided. Out of over 1300 accepted papers at EMNLP 2021, 723 had URLs to a repository containing the code required to run their experiments. Our selected papers are provided below. To save space, we refer to each paper by its associated number in this list.

1. A Massively Multilingual Analysis of Cross-linguality in Shared Embedding Space (Jones et al., 2021)
2. Automatically Exposing Problems with Neural Dialog Models (Yu and Sagae, 2021)
3. Frustratingly Simple but Surprisingly Strong: Using Language-Independent Features for Zero-shot Cross-lingual Semantic Parsing (Yang et al., 2021)
4. Weakly-supervised Text Classification Based on Keyword Graph (Zhang et al., 2021)
5. ReasonBERT: Pre-trained to Reason with Distant Supervision (Deng et al., 2021)
6. StreamHover: Livestream Transcript Summarization and Annotation (Cho et al., 2021)
7. ValNorm Quantifies Semantics to Reveal Consistent Valence Biases Across Languages and Over Centuries (Toney and Caliskan, 2021)
8. Measuring Association Between Labels and Free-Text Rationales (Wiegreffe et al., 2021)

While random selection of the papers avoids inserting selection bias into our findings, it may increase the difficulty of achieving reproducibility due to lower familiarity with certain concepts. We allotted fixed, limited time and computation resources for each paper, and report whether we were able to reproduce the findings of the paper within our time and resource budget.

5 Results

In this section we describe the results obtained, and how they answer the following research questions (RQs):

RQ1 Has code availability for published NLP literature improved in recent years?

RQ2 Is code availability enough for reproducibility?

RQ3 What new guidelines can be used to support reproducibility?

5.1 Quantitative Analysis

To analyze trends in source code availability, we select five major NLP conferences (ACL, EMNLP, LREC, NAACL, and COLING), and scrape the ACL Anthology to obtain data associated with all papers published at those venues from 2016 until the time of writing this paper (early summer 2022). Out of these conferences, LREC (LREC Organizers, 2022) and COLING (COLING Organizers, 2022) have not yet formally emphasized *reproducibility* in their submission process. On the other hand, EMNLP (EMNLP Organizers, 2022), ACL (ACL-IJCNLP Organizers, 2022), and NAACL (NAACL Organizers, 2022) highlight several reproducibility guidelines for authors to consider during the submission process. Figure 1 illustrates trends in published papers with respect to the frequency of code availability (top) and the ratio of published papers with code (bottom) at all five conferences.

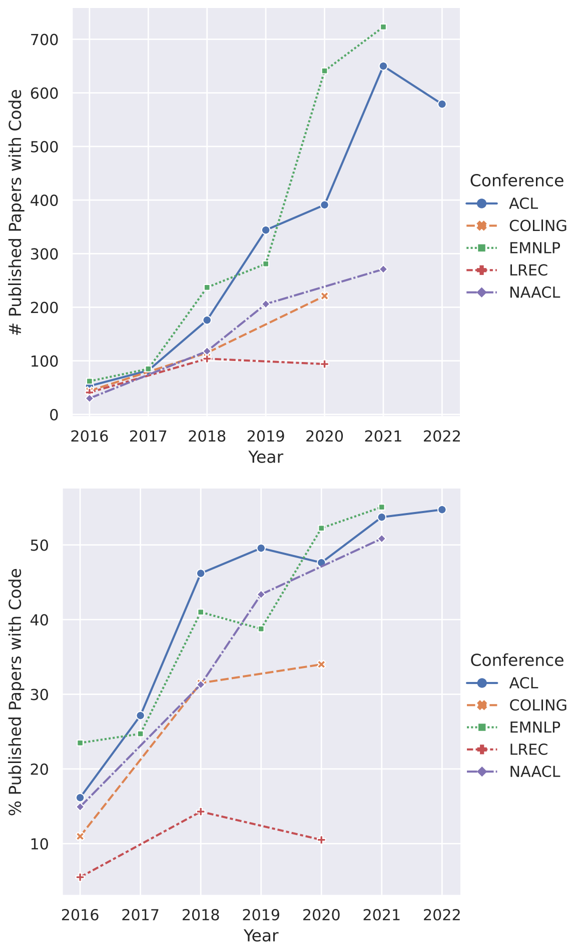

Figure 1: Trends of submissions with code over the last seven years. The top figure shows the absolute number of published papers with code per conference; the bottom figure shows the relative percentage of published papers with code per conference. LREC and COLING are organized every other year (even years). Data for the year 2022 was not available at the time of writing.

For perspective, Table 1 shows the total number of accepted papers at each of these conferences over the same time span.

RQ1: Figure 1 shows a positive trend in source code availability. Moreover, EMNLP, ACL, and NAACL appear to have the highest code submission ratio, which suggests that highlighting the importance of reproducibility throughout the submission process does lead to increased availability of source code. The surge of the number of published papers with code is especially noticeable for EMNLP 2020 and ACL 2021.

5.2 Reproducibility of Selected Studies

Conference	# Accepted Papers
ACL	5350
EMNLP	4810
LREC	2368
NAACL	1909
COLING	1436

Table 1: Total number of accepted papers at five major NLP conferences from 2016 to the present.

To understand how code availability impacts reproducibility, we randomly selected eight papers with code available from the EMNLP 2021 conference, as described in Section 4. This conference had the highest number of code submissions accompanying publications at the time of writing this paper. For each selected paper, we attempted to reproduce the reported results to the extent possible. If we encountered issues during this process, we tried to fix the issue to the best of our knowledge. If our attempts failed, we requested the authors' help through GitHub and email. We release self-contained environments that either reproduce the results or reproduce the error(s) we faced in this process for each paper.

Table 2 provides an overview of the availability of instructions, dependency specifications, and scripts used for training and evaluation. It also indicates whether we were able to reproduce the results for each paper. This checklist highlights the availability of materials that ultimately aid the reproducibility process. Given that one of the biggest reproducibility hurdles is getting the provided code to a running state, we expected this table to provide an estimate of the difficulty in reproducing the results of each selected paper.

Paper 1. The released source code for Paper 1 (Jones et al., 2021) contains a list of dependencies and full information on how to run the scripts provided. We found several syntax errors within the code released; fixing these errors took little effort. Unfortunately, the released code then terminated with a runtime error after executing for approximately one hour. We reported this issue to the authors through GitHub's tracking issue[1] four months prior to our camera-ready submission deadline, and followed up via email to the first author 45 days prior to the same deadline. The authors responded to the follow-up email but a solution had not been reached at the time of this writing (October 2022). Thus, our efforts failed to get the released source code to a running state.

Paper 2. The source code released for Pa-

[1] https://github.com/AlexJonesNLP/XLAnalysis5K/issues/1

Title	Inst.	Dep.	Scr.	Rep.
Paper 1 (Jones et al., 2021)	Yes	Yes	Yes	No
Paper 2 (Yu and Sagae, 2021)	No	No	No	No
Paper 3 (Yang et al., 2021)	Yes	Yes	Yes	No
Paper 4 (Zhang et al., 2021)	Yes	Yes	Yes	No
Paper 5 (Deng et al., 2021)	Yes	Yes	Yes	Yes
Paper 6 (Cho et al., 2021)	Yes	Yes	Yes	No
Paper 7 (Toney and Caliskan, 2021)	No	No	No	No
Paper 8 (Wiegreffe et al., 2021)	Yes	Yes	Yes	Yes

Table 2: Overall results of the reproducibility attempts discussed in this paper. Inst.=Instructions in form of README, Dep.=Specification of dependencies, Scr.=Training/evaluation scripts, Rep.= whether we were able to achieve reproducibility

per 2 (Yu and Sagae, 2021) contains no information or documentation describing how to achieve the reported results. We were unable to understand which files were used to achieve the results reported in the paper. We reached out to the authors five months prior to our camera-ready submission deadline through GitHub[2] to request further instructions, and followed up via email to the first author 45 days prior to the same deadline. However, we had not heard back from the authors at the time of this writing.

Paper 3. Paper 3 (Yang et al., 2021) requires downloading pre-trained embeddings to achieve the reported results. However, the embeddings are no longer available.[3] Additionally, we suspect that the scripts used for preprocessing the dataset are not available. We contacted the authors regarding this issue five months before the camera-ready submission deadline via GitHub.[4] We also sent a follow-up email to the first author 45 days prior to that deadline, but at the time of this writing we had not received a response.

Paper 4. According to the authors of Paper 4 (Zhang et al., 2021), their released source code requires multiple GPUs, and it is not possible to run on it on a single GPU. This requirement raises the entry barrier for assessing the reproducibility of this work. Fortunately, we had access to a multi-GPU workstation, so we were able to continue with our reproducibility analysis only to find two errors. First, there was a missing dependency, which was straightforward to fix. Second, we ran into a mismatched device error during the source code runtime, which originates from mishandling the device (CPU or GPU) used for the data or the model. Regardless of what device is used, if there is an operation between two tensors, they have to be on the same device. We reported this issue through GitHub four months before the camera-ready deadline for this paper.[5] Although we did not receive a response prior to the initial submission, we did hear back later after we followed up with the first author via email. The authors responded with a solution. Other users reported experiencing the same issue and that the solution appears to be working. Due to time constraints, we were unable to test the solution ourselves. Nonetheless, this case offers a good example of effective communication and collaboration to improve the reproducibility of the publication.

Paper 5. We were able to run the source code released for Paper 5 (Deng et al., 2021) without any issues. The source code included quick experiments with smaller data samples, which made the reproducibility assessment of this work easy and straightforward. Additionally, the authors provided clear and concise instructions on how to achieve the results reported in the paper. The training took minutes for each model, which would have made debugging easier if we had encountered issues.

Since we were able to successfully reproduce this paper, we measured the precision (*CV**) for the results of several models on the SQuAD dataset reported in Table 4 of Paper 5 (Deng et al., 2021). We present the results in the top portion of Table 3. We note that the reported results are from an average of five runs; however, to the best of our knowledge, the authors have not released the full results. We observe precision values ranging from 17.04 to 139.92. This is less than ideal, but it can be justified by the small size of the dataset and the random seed affecting the data order. We believe that running more experiments to increase the sample size would yield better precision. We mark this reproducibility attempt as successful.

[2] https://github.com/DianDYu/trigger/issues/1

[3] http://www.let.rug.nl/rikvannoord/DRS/embeddings/

[4] https://github.com/SALT-NLP/Multilingual-DRS-Semantic-Parsing/issues/1

[5] https://github.com/zhanglu-cst/ClassKG/issues/5

Paper	Model	Reported	Ours	Mean	Unbiased St. Dev.	*CV** ↓
5	BERT	9.9	5.78	7.84	3.64	52.25
	ReasonBERT$_R$	41.3	34.79	38.04	5.76	17.04
	ReasonBERT$_B$	33.2	5.81	19.50	24.26	139.92
	SSPT	10.8	4.92	7.86	5.20	74.51
	SpanBERT	15.7	10.07	12.88	4.98	43.53
8	E-SNLI	90.52	87.72	89.12	2.48	3.13

Table 3: Partial reproducibility results for Papers 5 (Deng et al., 2021, Table 4) and 8 (Wiegreffe et al., 2021, Table 3). Performance for Paper 5 (Deng et al., 2021) was measured as SQuAD dataset F_1 using a sample size of 16, and precision (*CV**) scores are averaged across five runs. Performance for Paper 8 (Wiegreffe et al., 2021) was measured as accuracy of the trained self-rationalizing model (I→OR). Lower CV^* is better.

Paper 6. The source code for Paper 6 (Cho et al., 2021) contains a full list of dependencies and instructions on how to run the code. As part of their evaluation, the source code used *pyrouge* to calculate the ROUGE metric (Lin, 2004). Even though we were able to train a model according to the instructions, the final step of evaluation failed with a runtime error due to the missing installation of ROUGE. Following instructions provided by the *pyrouge* Python release package was not possible due to an unavailable (dead) URL that was supposed to explain how to install ROUGE.[6] Even after finding the instructions included in the main GitHub repository for *pyrouge*, we were not able to get it to a working state. We suspect this package is no longer being maintained as it had not been updated for more than three years at the time of writing. Furthermore, this issue was already reported by others in April 2021.[7] We contacted the authors via email offering our collaboration to migrate the source code to use SacreROUGE (Deutsch and Roth, 2020) 45 days prior to the camera-ready deadline for this manuscript but did not receive a response.

Paper 7. The source code released with Paper 7 (Toney and Caliskan, 2021) is a collection of functions within a Python script. We were unable to determine which function(s) should be run for which experiment by inspecting the script. Without having access to the specific scripts used to run the experiments reported in the paper (or more documentation), we could not continue our work. We reached out to the authors by submitting an issue over GitHub four months prior to the camera-ready deadline for this paper,[8] and sent a follow-up email to the first author 45 days prior to the same deadline. At the time of this writing, we had not received an answer.

Paper 8. Paper 8 (Wiegreffe et al., 2021) raised our concerns about the hardware requirements for training and evaluation, as it uses T5 (Raffel et al., 2020)—one of the largest available pretrained Transformer models. Fortunately, the paper used T5 base, one of the smaller variants, and we were able to train and evaluate this model using a GPU with 24GB memory available. Unlike Paper 5 (Deng et al., 2021), this work did not contain a set of experiments using only a portion of the dataset. Given that training for 200 epochs (the authors' instructions) was beyond our allotted computing budget, we resorted to reducing the number of training epochs to one. This reduced the training time to less than an hour. Despite this reduction, our results were still close to those originally reported. We present these results in the bottom portion of Table 3.

RQ2: Unfortunately, our results show that code availability is not enough for reproducing the results present in published literature. Out of eight papers with released source code, we were only able to run two without issues. Furthermore, even though we made our best attempts to fix the issues with the others (including contacting the original authors), we were not successful in doing so.

5.3 Guidelines for Future Reproducibility

To determine what new guidelines can be introduced to improve the state of reproducibility in NLP, we first categorize the issues we found and check whether the existing guidelines cover them. The primary problem of Paper 2 (Yu and Sagae, 2021) and Paper 7 (Toney and Caliskan, 2021) was missing files, scripts, and instructions used to generate the reported results. Current reproducibility

[6] https://pypi.org/project/pyrouge/

[7] https://github.com/bheinzerling/pyrouge/issues/38

[8] https://github.com/autumntoney/ValNorm/issues/1

guidelines already address this problem. The ML Code Completeness checklist (Stojnic, 2022) highlights the importance of dependencies, code used for training and evaluation, and a README file accompanied by the instructions. We found that papers present training and evaluation scripts in many unique ways. This hinders the understandability of the code (*e.g.,* which script achieves which result, or what is the correct order of operations). We believe this problem could be addressed by recommending that authors include explicit scripts to generate each result reported in the paper.

Dependency on external resources was the main issue with Paper 3 (Yang et al., 2021), Paper 4 (Zhang et al., 2021), and Paper 6 (Cho et al., 2021). Paper 3 (Yang et al., 2021) used a pre-trained embedding file that was no longer available for download. Paper 4 (Zhang et al., 2021) and Paper 6 (Cho et al., 2021) had missing and broken dependencies, respectively. Dependencies introduce variability over time, and may become broken as packages cease to be maintained. Simply listing dependencies, even with exact versions (which may become broken or inaccessible in the future), is not adequate to ensure long term reproducibility. Instead, *self-contained artifacts* such as Docker containers and Virtual Machine (VM) images can recreate an executing environment with high fidelity without relying on external resources (*e.g.,* files or URLs). The use of virtual environments is already mentioned in the ML Code Completeness Checklist (Stojnic, 2022). The NAACL reproducibility track (Deutsch et al., 2022) also focuses on model verification using Docker containers (but the details have not been released at the time of writing). Outside of NLP, the NeurIPS Code and Data Submission Guidelines (NeurIPS, 2022) suggest the submitted codes should be self-contained and executable. Regardless, we believe reproducibility standards need to prioritize releasing self-contained environments. This shift would reduce the workload near submission deadlines while helping authors to document and record their work throughout development.

Except for Paper 5 (Deng et al., 2021) and Paper 8 (Wiegreffe et al., 2021), we encountered issues requiring the authors' assistance (*e.g.,* syntax and runtime errors). Aside from one case, our attempts to communicate with the authors were not successful. We understand that authors may not be available after their work is published, and that there are no guidelines regarding support of published research. Instead, this further strengthens our recommendation that future conferences require self-contained artifacts. We also recommend that they provide a venue to evaluate such artifacts at the time of publication, as performed in other fields of computer science (Wadler and Drossopoulou, 2021; Jhala and Dillig, 2022; Ali and Vitek, 2022; Falsafi et al., 2022).

On the positive side, Paper 5 (Deng et al., 2021) eased our reproducibility attempt through the inclusion of small-scale experiments. Often, the resources available for assessing reproducibility are limited compared to the original study. Therefore, unique hardware requirements and compute-intensive methods raise the barrier for reproducibility assessments. We recommend including limited experiments that are able to run on commodity hardware and with modest time requirements.

RQ3: Given the empirical results provided in the previous section, we believe the following guidelines would help future reproducibility:

1. Include small scale experiments.
2. Include and document explicit scripts to generate each result in the paper.
3. Release executable self-contained artifacts.
4. Require (and evaluate) artifacts, not source code.

6 Discussion

It is encouraging to observe the recent upward trend in releasing source code across the broad NLP community, thanks to the recent focus on reproducibility. Unsurprisingly, conferences not promoting reproducibility standards have fewer submissions with included code. Therefore, we encourage all conferences to include such standards in their submission process.

Unfortunately, our results suggest that submitting the code alone does not seem to be enough, as the released code does not meet a minimum requirement for reproducibility, defined as achieving the results reported in the paper using the provided source code. We believe it is time to improve reproducibility standards to address the concerns we raise in regards to quality of the released source code. In particular, we strongly suggest shifting the focus *from source code to research artifacts* which include (1) a self-contained runtime environment (*e.g.,* a Docker container or a VM image) with scripts for achieving every single result reported in the paper, (2) smaller experiments to quickly validate the integrity of the artifact, and (3) extensive documentation explaining how to run the code. We took a first step in this direction by packaging all the results in the paper in their own artifact, and releasing it with this paper. These artifacts are available through Zenodo (Arvan et al., 2022).

Reproducibility is a desired attribute for solutions in natural language processing, but it comes with a cost. There may exist cases in which the required conditions for reproducing the results are not practical. Defining what is practical, of course, depends on the problem at hand. For example, reproducing a named entity classifier that requires using the same hardware may not be considered practical. This phenomenon is quite similar to the bias-variance tradeoff. Bias-variance, a property of statistical and machine learning models, suggests that the variance of the parameter estimated across samples can be reduced by increasing the bias. A dilemma exists when trying to minimize these two sources of error simultaneously. We have a dilemma when it comes to assessing the reproducibility of results. Many attempts have focused on controlling all the variables. Yet, while they have their use cases, their complexity makes them less viable. Perhaps a better alternative is to reduce the emphasis on the top-performing results and utilize techniques that attempt to aggregate and report the results of a set of experiments.

7 Conclusions

In this paper, we examined the trends of source code availability at major NLP conferences in recent years. We also performed a reproducibility study on eight randomly selected papers. After achieving a 25% success rate, we draw attention to the primary causes of irreproducibility of the selected papers. Based on the findings from our case study, we close by providing recommendations for improving the state of reproducibility in NLP. Researchers concerned about the added workload should come to terms with the fact that accessibility and reproducibility are fundamental, rather than auxiliary, cornerstones of strong research. It is also advantageous to researchers: following the recommendations proposed in this paper is anticipated to prolong and future-proof work, minimizing the required support in the long term.

Limitations

We have done our best to provide logical and step-by-step reasoning to describe our work. However, we have also identified a few limitations. First, some papers may have released their source code but not included it in their submission process, leading to inaccuracies in the trends computed based on data scraped from the ACL Anthology portal. Additionally, our reproducibility analysis only examines a small number of recent papers. To avoid the risk of selection bias, we selected the papers randomly. We believe our results were obtained from a representative sample. We have included additional information for each selected paper in Appendix A.

We failed to reproduce the results for six papers (75%), which may be attributed to our own lack of expertise. We allotted a similar amount of effort and time to each paper. We used this time to fix the issues and contact the authors in case we were not able to do so. In theory, we could have devoted a large enough amount of effort to each paper to reproduce it successfully. However, in practice and without help from the authors, it is unclear how long this would take, and we believe that such an approach would amount to a complete reimplementation of the original paper, which is outside of the scope of this work.

Finally, in our recommendations we emphasized the importance of self-contained environments. However, this may not be achievable in every case. For instance, in the healthcare domain, datasets often contain private and sensitive information. Releasing such datasets is not an option, and thus achieving the same degree of reproducibility is not possible in those fields.

Ethical Considerations

In this work, we make informed recommendations designed to improve the state of research reproducibility in natural language processing. When research is easily reproducible, its access is extended to a broader community of researchers. However, requiring authors to provide self-contained environments may also have unwanted side effects. In particular, researchers without high-bandwidth internet connections may find it difficult to upload gigabytes of data as part of their paper submission. We believe that providing alternative and delayed forms of submitting self-contained environments should eliminate this issue.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful feedback, which was incorporated into the final version of this paper. This work was partially supported by equipment purchased using the University of Illinois Chicago's Provost's Graduate Research Award, and by startup grants from the University of Illinois Chicago.

References

AAAI. 2022. AAAI Reproducibility Checklist.

ACL. 2022a. ACL Responsible NLP Research.

ACL. 2022b. ACL Responsible NLP Research.

ACL-IJCNLP Organizers. 2022. ACL-IJCNLP Call for Papers.

ACM. 2022. ACM Artifact Review and Badging.

Karim Ali and Jan Vitek, editors. 2022. *36th European Conference on Object-Oriented Programming, ECOOP 2022, June 6-10, 2022, Berlin, Germany*, volume 222 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Mohammad Arvan, Luís Pina, and Natalie Parde. 2022. Artifacts of Reproducibility in Computational Linguistics: Is Source Code Enough?

Anya Belz, Maja Popovic, and Simon Mille. 2022. Quantified reproducibility assessment of NLP results. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16–28, Dublin, Ireland. Association for Computational Linguistics.

Anya Belz, Anastasia Shimorina, Shubham Agarwal, and Ehud Reiter. 2021. The ReproGen shared task on reproducibility of human evaluations in NLG: overview and results. In *Proceedings of the 14th International Conference on Natural Language Generation, INLG 2021, Aberdeen, Scotland, UK, 20-24 September, 2021*, pages 249–258. Association for Computational Linguistics.

Sangwoo Cho, Franck Dernoncourt, Tim Ganter, Trung Bui, Nedim Lipka, Walter Chang, Hailin Jin, Jonathan Brandt, Hassan Foroosh, and Fei Liu. 2021. Streamhover: Livestream transcript summarization and annotation. In (Moens et al., 2021b), pages 6457–6474.

COLING Organizers. 2022. COLING Call for Papers.

Xiang Deng, Yu Su, Alyssa Lees, You Wu, Cong Yu, and Huan Sun. 2021. Reasonbert: Pre-trained to reason with distant supervision. In (Moens et al., 2021b), pages 6112–6127.

Daniel Deutsch, Yash Kumar Lal, Annie Louis, Pete Walsh, Jesse Dodge, and Niranjan Balasubramanian. 2022. 2022 North American Chapter of the Association for Computational Linguistics Reproducibility Track.

Daniel Deutsch and Dan Roth. 2020. SacreROUGE: An Open-Source Library for Using and Developing Summarization Evaluation Metrics. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 120–125, Online. Association for Computational Linguistics.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2185–2194. Association for Computational Linguistics.

EMNLP Organizers. 2022. EMNLP Call for Papers.

Babak Falsafi, Michael Ferdman, Shan Lu, and Thomas F. Wenisch, editors. 2022. *ASPLOS '22: 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Lausanne, Switzerland, 28 February 2022 - 4 March 2022*. ACM.

Odd Erik Gundersen and Sigbjørn Kjensmo. 2018. State of the art: Reproducibility in artificial intelligence. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1644–1651. AAAI Press.

JCGM. 2012. JCGM 200: 2012 International Vocabulary of Metrology: Basic and General Concepts and Associated Terms (VIM). *The Joint Committee for Guides in Metrology and The Bureau International des Poids et Mesures: Paris, France*.

Ranjit Jhala and Isil Dillig, editors. 2022. *PLDI '22: 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation, San Diego, CA, USA, June 13 - 17, 2022*. ACM.

Alexander Jones, William Yang Wang, and Kyle Mahowald. 2021. A massively multilingual analysis of cross-linguality in shared embedding space. In (Moens et al., 2021b), pages 5833–5847.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

LREC Organizers. 2022. LREC Call for Papers.

Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors. 2021a. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic.

Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors. 2021b. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics.

NAACL Organizers. 2022. NAACL Call for Papers.

Nature. 2022. Nature's Reporting standards and availability of data, materials, code and protocols.

NeurIPS. 2022. NeurIPS 2022 Code and Data Submission Guidelines.

Babatunde Kazeem Olorisade, Pearl Brereton, and Peter Andras. 2017. Reproducibility of studies on text mining for citation screening in systematic reviews: Evaluation and checklist. *J. Biomed. Informatics*, 73:1–13.

Joelle Pineau. 2019. ML Reproducibility Checklist.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Hugo Larochelle. 2021. Improving reproducibility in machine learning research(a report from the neurips 2019 reproducibility program). *J. Mach. Learn. Res.*, 22:164:1–164:20.

Edward Raff. 2019. A step toward quantifying independently reproducible machine learning research. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5486–5496.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Anna Rogers, Timothy Baldwin, and Kobi Leins. 2021. 'just what do you think you're doing, dave?' A checklist for responsible data use in NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 4821–4833. Association for Computational Linguistics.

Nicolas P. Rougier, Konrad Hinsen, Frédéric Alexandre, Thomas Arildsen, Lorena A. Barba, Fabien C. Y. Benureau, C. Titus Brown, Pierre de Buyl, Ozan Caglayan, Andrew P. Davison, Marc-André Delsuc, Georgios Detorakis, Alexandra K. Diem, Damien Drix, Pierre Enel, Benoît Girard, Olivia Guest, Matt G. Hall, Rafael Neto Henriques, Xavier Hinaut, Kamil S. Jaron, Mehdi Khamassi, Almar Klein, Tiina Manninen, Pietro Marchesi, Dan McGlinn, Christoph Metzner, Owen L. Petchey, Hans Ekkehard Plesser, Timothée Poisot, Karthik Ram, Yoav Ram, Etienne B. Roesch, Cyrille Rossant, Vahid Rostami, Aaron Shifman, Joseph Stachelek, Marcel Stimberg, Frank Stollmeier, Federico Vaggi, Guillaume Viejo, Julien Vitay, Anya E. Vostinar, Roman Yurchak, and Tiziano Zito. 2017. Sustainable computational science: the rescience initiative. *PeerJ Comput. Sci.*, 3:e142.

Patrick D Schloss. 2018. Identifying and overcoming threats to reproducibility, replicability, robustness, and generalizability in microbiome research. *MBio*, 9(3):e00525–18.

Koustuv Sinha, Jesse Dodge, Sasha Luccioni, Jessica Forde, Sharath Chandra Raparthy, François Mercier, Joelle Pineau, and Robert Stojnic. 2021. ML Reproducibility Challenge 2021.

Robert Stojnic. 2022. ML Code Completeness Checklist.

Autumn Toney and Aylin Caliskan. 2021. Valnorm quantifies semantics to reveal consistent valence biases across languages and over centuries. In (Moens et al., 2021b), pages 7203–7218.

Philip Wadler and Sophia Drossopoulou, editors. 2021. volume 5. Association for Computing Machinery.

Kirstie Whitaker. 2017. Showing your working: a how to guide to reproducible research. *"slideshare"*.

Sarah Wiegreffe, Ana Marasovic, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. In (Moens et al., 2021b), pages 10266–10284.

Martijn Wieling, Josine Rawee, and Gertjan van Noord. 2018. Reproducibility in computational linguistics: Are we willing to share? *Comput. Linguistics*, 44(4).

Jingfeng Yang, Federico Fancellu, Bonnie Webber, and Diyi Yang. 2021. Frustratingly simple but surprisingly strong: Using language-independent features for zero-shot cross-lingual semantic parsing. In (Moens et al., 2021b), pages 5848–5856.

Dian Yu and Kenji Sagae. 2021. Automatically exposing problems with neural dialog models. In (Moens et al., 2021b), pages 456–470.

Lu Zhang, Jiandong Ding, Yi Xu, Yingyao Liu, and Shuigeng Zhou. 2021. Weakly-supervised text classification based on keyword graph. In (Moens et al., 2021b), pages 2803–2813.

Title	Stars	Forks	Issues	DSLU
Paper 1 (Jones et al., 2021)	1	1	0	47
Paper 2 (Yu and Sagae, 2021)	0	1	1	132
Paper 3 (Yang et al., 2021)	3	1	1	128
Paper 4 (Zhang et al., 2021)	15	4	1	22
Paper 5 (Deng et al., 2021)	23	3	1	26
Paper 6 (Cho et al., 2021)	5	1	0	106
Paper 7 (Toney and Caliskan, 2021)	3	2	0	379
Paper 8 (Wiegreffe et al., 2021)	5	1	0	101
EMNLP 2021	234.44 ± 1889	53.15 ± 456	7.58 ± 59	72.7 ± 77

Table 4: GitHub stars, forks, open issues, and days since the last update (DSLU) for the selected papers.

A Selected Paper GitHub Repository Information

In addition to tracking elements of the code releases themselves (*e.g.,* whether they included instructions, specified their dependencies, or provided their training and evaluation scripts), we also recorded broader metrics associated with the repositories for our included papers. In Table 4 we present the number of GitHub stars, forks, open issues, and days since last update (at the time of writing) for each paper. In the last row, we also provide this same information in aggregate form across all EMNLP 2021 papers with linked GitHub repositories.