NLP Reproducibility For All: Understanding Experiences of Beginners

Shane Storks   **Keunwoo Peter Yu**   **Ziqiao Ma**   **Joyce Chai**

Computer Science and Engineering Division, University of Michigan

`{sstorks, kpyu, marstin, chaijy}@umich.edu`

Abstract

As natural language processing (NLP) has recently seen an unprecedented level of excitement, and more people are eager to enter the field, it is unclear whether current research reproducibility efforts are sufficient for this group of *beginners* to apply the latest developments. To understand their needs, we conducted a study with 93 students in an introductory NLP course, where students reproduced the results of recent NLP papers. Surprisingly, we find that their programming skill and comprehension of research papers have a limited impact on their effort spent completing the exercise. Instead, we find accessibility efforts by research authors to be the key to success, including complete documentation, better coding practice, and easier access to data files. Going forward, we recommend that NLP researchers pay close attention to these simple aspects of open-sourcing their work, and use insights from beginners' feedback to provide actionable ideas on how to better support them.

1   Introduction

As natural language processing (NLP) research continues to grab public attention and excitement, it becomes increasingly important for it to be accessible to a broad audience. While the research community works to democratize NLP, it remains unclear whether *beginners* in the field can easily apply the latest developments. How easy is it for them to reproduce experimental results? Will their programming background or comprehension of papers play a role? What key elements affect their experience here?

To address these questions, we conducted a controlled user study in an introductory NLP class. We first identified and successfully reproduced the results of three recent NLP publications ourselves. We surveyed students on their background in machine learning and programming, then divided them into three skill level groups based on

the survey results. Each group was asked to reproduce the results of the three papers. As students conducted experiments, they tracked their own efforts, while their computational resource usage was logged by a shared high-performance computing system. After conducting the experiments, students answered questions and provided feedback about their experiences.

Our results show that beginners' technical skill level and comprehension of the papers play only a small part in their experience in reproducing the results, and we observed a strikingly wide range of time spent on the exercise regardless of these user-specific factors. Meanwhile, we show that reproducibility efforts by paper authors make a much larger impact on user experience, and based on direct feedback from these beginners, we find that they encounter a range of roadblocks related to the documentation and ease of use of open-sourced materials. These findings shed light on the extra steps NLP researchers can take to make state-of-the-art technology more accessible to beginners, an important direction to continue democratizing NLP to its growing audience. To this end, we make several concrete recommendations in Section 5 on how the NLP research community can further improve accessibility to beginners.

2   Related Work

Amidst growing concern of a reproducibility crisis across many scientific disciplines (Baker, 2016), recent years have seen an increasing amount of work studying trends and proposing guidelines to improve research reproducibility in NLP and neighboring disciplines (Arabas et al., 2014; Rozier and Rozier, 2014; Sethi and Gil, 2016; Henderson et al., 2018; Crane, 2018; Cohen et al., 2018; Tatman et al., 2018; Dodge et al., 2019; Pineau, 2020; Pineau et al., 2021; Rogers et al., 2021; Belz, 2022). As part of these efforts, the Association for Computational Linguistics (ACL) has also adopted author

checklists for reproducibility and responsible research practice.

As a prerequisite for reproducibility in NLP, prior work has studied the availability of code and data at scale (Wieling et al., 2018). Where code and data are available, large-scale and multi-test studies have assessed the reproducibility of results (António Rodrigues et al., 2020; Branco et al., 2020). Various venues have invited individual reproductions and replications of work in machine learning and NLP, including the Reproducibility, Inexplicability, and Generalizability of Results Workshop (Arguello et al., 2016) in information retrieval, the Reproducibility in ML workshops[1] at the International Conference on Machine Learning (ICML) and International Conference on Learning Representations (ICLR), and the ML Reproducibility Challenges[2] at ICLR and Neural Information Processing Systems (NeurIPS). Belz et al. (2021) reviewed and aggregated many of these efforts in NLP, finding only up to 14% of experiments led to accurately reproduced results. Unlike prior work, we are not focused on the accuracy of results, but rather NLP beginners' experiences in reproducing results that are pre-verified to be reproducible.

3 Methodology

Next, we introduce the methodology for our study, including the steps taken for data collection, and the variables of interest we study in our analysis of beginners' experience reproducing NLP results.

3.1 Data Collection

Data collection for this study consisted of several steps, outlined below.

3.1.1 Pre-Survey

First, we conducted a pre-survey on students' backgrounds and their understanding of course material. We pose our questions based on the categories proposed by Feigenspan et al. (2012) to measure programming experience, and ask students to provide informed consent to use their survey responses for the purpose of this study. The full set of pre-survey questions can be found in Appendix D.

3.1.2 Paper Selection & Expert Reproduction

Next, we carefully selected a small number of papers in recent ACL conferences, verifying that we could reproduce their results in a comparable, reasonable amount of time and effort using a single GPU[3] from an institution-provided computing cluster managed with Slurm.[4] In line with the definition of reproducibility from Rougier et al. (2017), we used the code and data published by the paper authors to attempt to reproduce results, rather than re-implementing models and algorithms. Out of 24 papers considered from 2018 through 2022, we could accurately reproduce results within these bounds from only 3 papers. Common reasons for failure to reproduce results included long model training time requirements, incompatibility of code bases with our computing platform, incomplete documentation, and discrepancies between reproduced results and those reported in papers. Selected reproducible papers are identified in Table 1,[5] while the selection process is fully documented in Appendix C. Selected papers come from one track (Sentence-Level Semantics and Textual Inference), minimizing the impact of varying subject matter complexity among different topic areas.

It is important to note that the goal of this research is toward user reproducibility *experience*, which is different from previous works focusing on the accurate reproducibility of results. Therefore, instead of having many papers to conduct the experiments, we chose to control the study by selecting a small number of comparable papers with different characteristics in their released code. As each paper was reproduced by a group of students with different skill levels, we expected to gather sufficient statistics to identify common trends in responses to these characteristics.

3.1.3 Reproduction Process

In reproducing results themselves, students were required to use the same GPU computing resources within our university's shared cluster to control for the potential effect of computing infrastructure on results. Students used Slurm to request and acquire sole control of resources, and all resource utilization was automatically tracked by this centralized platform. To further control the impact of our specific computing environment, we ensured students' familiarity with the platform through an earlier assignment to implement and run state-of-

---

the-art NLP models on the platform. Each student was assigned to reproduce results from one of the selected papers (A, B, or C); more information on paper assignment is provided in Section 3.2.2. While reproducing experiments from their assigned paper, students tracked time spent setting up the code base, as directed in a homework assignment associated with this study.[6]

3.1.4 Post-Survey

After reproducing the results of each paper, students completed a survey about their experience. Here, we asked questions about their comprehension of the paper, time spent and difficulties encountered reproducing the results, and general outlooks on the experiment, including what helped and blocked their ability to reproduce results. Post-survey questions are listed in Appendix F.

3.2 Analysis of User Experience

Next, we introduce the key factors in interpreting our collected data, and how they may be characterized. Specifically, we consider a student's experience reproducing the results of their assigned paper as a dependent variable, and consider three types of independent variables: students' skill level, students' comprehension of a paper, and paper authors' specific efforts toward making results reproducible. Our goal is to understand how each of these variables can impact a beginner's experience in reproducing NLP results.

3.2.1 Defining User Reproducibility Experience

In this reproducibility study, we aim to understand a beginner's *experience* in reproducing results. We characterize this by students' **time spent** and **reported difficulty** to reproduce results.

Setup time and runtime. Time spent to reproduce results is divided into two phases:

1. **Setup time:** Downloading and setting up the code, dataset, and external dependencies.
2. **Runtime:** Training and evaluating systems.

System setup time is self-reported in the post-survey, while runtime is extracted from the centralized Slurm system using its resource usage tracking feature, which accurately reports the total GPU usage time per student. As such, runtime may include extra time a student may have spent for trial and

Paper	Reference	Setup	Runtime
A	Zhou et al. (2021)	2 hrs.	0.5 hr.
B	Donatelli et al. (2021)	2 hrs.	3 hrs.
C	Gupta et al. (2020)	2 hrs.	2 hrs.

Table 1: Selected papers for the study, and research team's code setup time and runtime[7] rounded to the nearest half hour. All papers are from the *Sentence-Level Semantics and Textual Inference* area in ACL venues.

error, including requesting extra GPUs that went unused. Calculating runtime this way is suitable for our purpose, as management of hardware resources could pose a real barrier to NLP beginners. As runtime varies significantly by paper, we quantify it by percent error from the research team's runtime when reproducing the same result. These variables can provide indirect yet objective measures on how much a student struggled with the experiment compared to other beginners and experts.

Difficulty ratings. For a more direct measure of students' experience, we also considered student ratings for difficulty encountered in each step of the experiment (on a scale from 1-5, 5 being most difficult):

1. **Downloading source code**, which requires cloning one or more GitHub repositories.
2. **Downloading data**, which may be hosted somewhere different than the code.
3. **Setting up the code base**, including installing external dependencies or pre-trained models.
4. **Data preprocessing**, which may entail running scripts or manual adjustment.
5. **System training**, which may require hyperparameter search or be informed by a preselected hyperparameter configuration.
6. **System evaluation**, where evaluation metrics directly comparable to the paper's must be calculated and reported.

3.2.2 Defining Skill Level

We may expect that a student's skill level or technical background may have an impact on their experience. As such, we collected data about students' programming background and understanding of NLP coursework in the pre-survey. To characterize student skill level, four variables are extracted from their responses:

1. **Python experience** (years)
2. **PyTorch experience** (years)

---

[6]Instructions given to students are listed in Appendix E.

[7]Calculation of setup time and runtime described in Section 3.2.1.

10201

Paper	Nov.	Int.	Adv.	Total
A	12	11	11	34
B	10	10	10	30
C	10	9	10	29

Table 2: Distribution of assigned papers across skill level groups (novice, intermediate, and advanced).

3. **LSTM understanding** (1-5 from worst to best understanding)

4. **Transformer understanding** (1-5 from worst to best understanding)

All skill level factors are self-reported. We focus on Python and PyTorch as these are commonly used in NLP research, including all selected papers. As such, knowledge of them may most directly transfer to reproducing NLP results. Meanwhile, students' understanding of LSTMs (Hochreiter and Schmidhuber, 1997) and transformers (Vaswani et al., 2017) is self-reported in the pre-survey based on related past homework assignments requiring students to implement them in PyTorch. This hands-on experience could contribute to their ability to reproduce results from the selected papers, each of which applied transformer-based models. Accounting for these factors equally, we divide the 93 study participants into 3 skill level groups as close to equal size as possible (considering ties): *novice*, *intermediate*, and *advanced*. As shown in Table 2, papers were distributed mostly uniformly within each skill level.[8]

3.2.3 Defining Comprehension

Meanwhile, we might expect that a student's comprehension of a specific paper could also contribute to their ability to reproduce its results. To measure students' comprehension of a paper objectively, we carefully designed a set of four-way multiple-choice questions about the key aspects of each work. Specifically, we asked about each paper's:

1. **Motivation**: Prior limitations in related work addressed by the paper.
2. **Problem Definition**: Target task details.
3. **Approaches**: Inputs and outputs of reproduced system.
4. **Implementation**: Matching of a process described in the paper to a file in the code.
5. **Results**: Evaluation criteria details.
6. **Conclusion**: Implications of results.

Students answered these questions in the post-survey. Questions were posed in such a way that the answers could not be found directly in the paper or code. While the specific question and answers of course varied by paper, their nature was standard, enabling us to consistently characterize students' comprehension of a paper. Together, answering these questions correctly implies a comprehensive understanding of the work which may be supported not just by reading the paper, but also working hands-on with the code. As such, we measure comprehension of the paper by students' accuracy on these questions as a whole, and can even use their correctness on specific questions to represent comprehension of specific aspects. The specific comprehension questions we asked students are listed in Appendix G.

3.2.4 Defining Author Reproducibility Efforts

A strong source of guidance for reproducibility in NLP is the ACL Reproducibility Checklist (ACLRC)[9] which authors must complete in order to submit manuscripts to many ACL publication venues.[10] While the items listed on the ACLRC are all important efforts for the reproducibility of NLP research in general, we would like to understand which of them are particularly important for beginners' success.

This is not straightforward for a few reasons. First, whether an item on the checklist is satisfied is subjective, as a paper's code release may provide some information pertaining to an item, but the degree to which it is easy to find and understand may vary. Second, the reproducibility of some papers may benefit from efforts toward certain items more than others. For example, if a paper entails a long model training time, reporting the expected training time may be especially beneficial for users of pay-per-use computing environments. Lastly, the degree to which a reproducibility effort is found helpful by users can be subjective. For example, one may find a reproducibility effort helpful just by its existence regardless of its quality, e.g., an unclear hyperparameter configuration buried in the code, while others may prioritize high quality.

For the most complete understanding of beginners' experiences, we make no restrictions along

---

[8]Subject consent caused minor non-uniformity in assignment distributions.

[9]See Appendix A for the full checklist.

[10]More recently, some items of this checklist have been adapted into the more comprehensive Responsible NLP Research checklist used at some venues (Rogers et al., 2021). We compare these checklists in Section 5.

Skill Level Factor	ρ (time)	ρ (diff.)
Python Experience (Years)	-0.291	-0.230
PyTorch Experience (Years)	-0.251	-0.259
LSTM Understanding (1-5)	**-0.430**	**-0.396**
Transformer Understanding (1-5)	-0.317	-0.338

Table 3: Spearman correlation coefficients for how well self-reported code setup time and difficulty is predicted by skill level factors, including PyTorch and PyTorch experience, and self-reported understanding of NLP models covered in course homework assignments. All correlations are statistically significant ($p < 0.05$); strongest correlations for each dependent variable in bold.

Figure 1: System setup time (minutes) and runtime (percent difference from research team's runtime) by skill level (novice, intermediate, or advanced).

these lines. For each paper, students selected items from the ACLRC that they specifically found to be most helpful toward reproducing the results of their assigned paper.[11] We use their responses to characterize each paper's efforts toward reproducibility.

4 Results & Findings

Here, we investigate how aspects of students' background and understanding of papers, as well as reproducibility efforts by the authors of papers, affected students' experience in reproducing experimental results.[12] Following prior work, we first verify the accuracy of the results. Students on average produced results with a relative error within 3% accuracy from reported results regardless of their skill level or assigned paper.[13] This is expected, as we verified that all papers' results can be accurately reproduced.

Next, we systematically investigate the experience students had reproducing results from the papers. We analyze their experience from three perspectives: students' skill level (Section 3.2.2), students' comprehension of their assigned paper (Section 3.2.3), and paper authors' efforts toward making their results more reproducible (Section 3.2.4).

4.1 Student Skill Level

First, we examine the relationship of a student's skill level with their experience, specifically the time taken to set up and run their experiment, as well as their opinion on how difficult it was.

Relationship with time. Figure 1 shows the distribution of setup times and runtimes reported by students. We observe a striking variation in setup time across all skill levels, from under an hour to
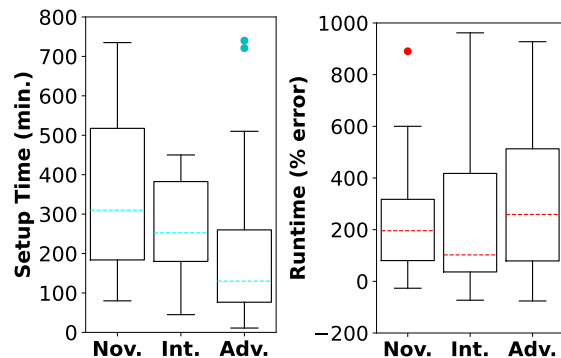
nearly 30 hours. As skill level increases, we observe that the median and minimum setup time, as well as the overall range of setup times, marginally decrease. To examine how factors used to assign their skill level contribute to setup time, we calculate the Spearman correlation between each skill level factor and setup time in the second column of Table 3. Indeed, we observe a significant correlation, with understanding of the homework assignment on LSTMs having the strongest negative association with setup time. As this assignment required implementing and training language models in PyTorch, this suggests that these hands-on NLP skills may save students' time when reproducing NLP results. However, if we interpret ρ^2 as a coefficient of determination, skill level factors explain only up to $\rho^2 = 18.5\%$ of variance in setup time.[14] The large overlap in the setup time distribution between skill levels further suggests that there are more factors at play here. Meanwhile, we see no clear differences in runtimes based on skill level, as each paper should have a consistent required runtime to train and evaluate models.

Relationship with difficulty. For a more direct measure of students' experience, Figure 2 summarizes student ratings for the difficulty of each step of the experiment. For most steps of the experiment, more novice students reported slightly more difficulty. Students found code setup, data preprocessing, and system training to be the most difficult steps, and we observed a significant decrease in difficulty with respect to skill level for code setup. This suggests a relationship between students' skill level and their reported code setup difficulty.

---

[11]Summary of student responses in Appendix B.3.

[12]Data analysis code shared at https://github.com/sled-group/NLP-Reproducibility-For-Beginners.

[13]See Appendix B.1 for more on accuracy of results.

[14]All variables have a high variance inflation factor, thus it is likely that they similarly contribute to a student's experience.
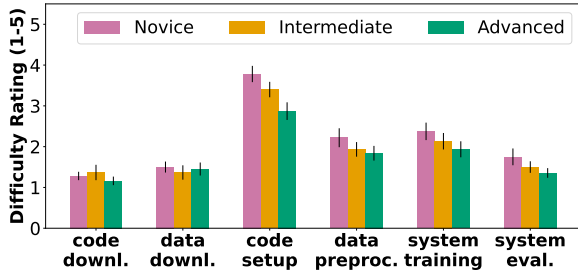
Figure 2: Mean reproducibility difficulty rating (1-5, 5 being most difficult) for each step of experiments: downloading the code and data, setting up the code, preprocessing data, and training and evaluating the system.

To understand this relationship, we calculate the Spearman correlation between each skill level factor and code setup difficulty rating, shown in the third column of Table 3. Again, we find that all skill level factors are significantly correlated, with LSTM understanding again having the strongest association with lower reported difficulty. Similarly, though, we observe a maximum $\rho^2 = 15.7\%$, suggesting that while some of students' reported difficulties may be explained by their skills, there is likely more to the story. Further, it remains to be seen what exactly makes novice students feel worse about the difficulty of experiments, or why the rating for code setup is leaning negative overall. The remainder of our analysis provides possible answers to these questions.

4.2 Student Comprehension of Paper

Knowledge in NLP is primarily transferred through research papers. A student's ability to absorb knowledge from their assigned paper may relate to their ability to reproduce its results. Here, we examine the relationship between their accuracies on paper comprehension questions in the post-survey and their experience, characterized by code setup time and difficulty rating, which exhibit the most significant variations across students.

As shown in Figure 3, we observed a wide range of accuracies on these questions, with no clear correlation to their reported setup time. There is not a significant Spearman correlation between question accuracy and setup time or difficulty rating, suggesting that a student's comprehension of the work is not associated with their experience in reproducing its results. This shows that even the clearest, most well understood paper may be difficult for beginners to engage with hands-on, and thus effective open-sourcing of code remains a separate and important issue to enable reproducibility.
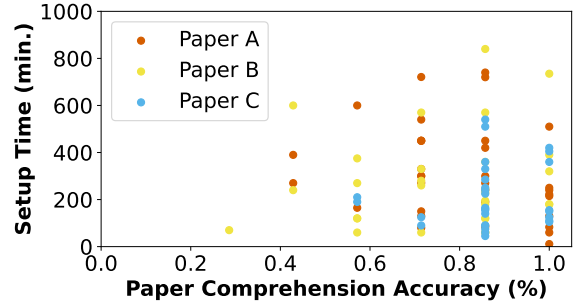

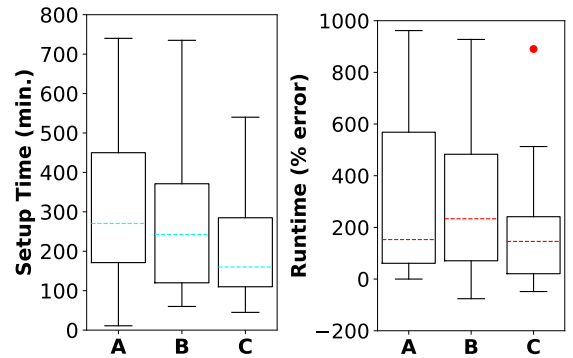Figure 3: Students' accuracy on comprehension questions versus time to set up the experiment.

Figure 4: System setup time (minutes) and runtime (percent difference from research team's runtime) by assigned paper.

4.3 Author Reproducibility Efforts

Our results so far are vexing, as they show that students encounter vastly different levels of roadblocks in reproducing results that are only somewhat attributed to their skills, and unrelated to their comprehension of the work. To enable a frustration-free experience for all beginners, it is essential to understand what causes this disparity. To investigate whether their experience instead depends on authors' specific efforts to make paper results reproducible, we examine the variations between the assigned papers in terms of time spent to reproduce results, as well as students' feedback on the experiments.

Relationship with time. First, we examine the distribution of setup time and runtime by assigned paper in Figure 4. A wide range of times is again observed for each paper, and the median setup time is consistently higher than the research staff's setup time of 2 hours, suggesting that compared to the experts on the research team, beginners are still learning to efficiently set up the code for an NLP experiment. Meanwhile, the median runtime for all papers is also higher than than those of the research team, and a wide range of runtimes are again ob-

served. This may suggest that across the board, beginners encountered issues that caused them to troubleshoot by running code repeatedly. Paper C's median and range of setup times, as well as its range of runtimes, were the smallest, possibly suggesting that the authors' efforts on reproducibility were especially helpful in enabling beginners to quickly reproduce their results.

To understand these observed differences across papers, we take a closer look at reproducibility efforts for each paper. After performing the experiment, students indicated in the post-survey which items from the ACLRC were most helpful in reproducing the results from their specific paper.[15] We analyze their responses with respect to setup time and runtime in Table 4 by performing a multiple linear regression over items from the ACL Reproducibility Checklist as predictors for students' code setup time and runtime.

We find several significant correlations between checklist items and students' setup time and runtime. Specifically, reporting of best model hyperparameters (Paper A), model description (Paper B), and dataset partition information (Paper C) are all positively correlated with setup time, with hyperparameter bounds also positively correlated with runtime for Paper A. This may suggest that students encountered issues related to these factors that contributed to longer observed setup times and runtimes. Meanwhile, reporting of model selection information was associated with faster runtimes for Paper B, suggesting this was well addressed and important for a successful reproduction.

This analysis provides detailed insights about how each paper differed in terms of serving the needs of beginners. Notably, the top-ranked reproducibility efforts can explain a relatively large amount of variance in setup time and runtime for some papers. For example, $R^2 = 62\%$ of variance in Paper C's setup time and 66% of variance in Paper B's runtime are explained by these efforts, providing some strong reasons for the wide ranges observed. This may suggest that these particular efforts are crucial for beginners' experience, and in general, that authors' reproducibility efforts can have a much larger impact on their experience than their technical skills or paper comprehension.

Paper	Top ACLRC Item, Setup Time	β	R^2
A	10. Best Hyperparameters	**4.24**	0.53
B	1. Model Description	**8.47**	0.15
C	14. Dataset Partition Info	**4.08**	0.62
All	1. Model Description	1.89	0.40

Paper	Top ACLRC Item, Runtime	β	R^2
A	9. Hyperparameter Bounds	**46.43**	0.17
B	11. Model Selection Strategy	**-13.20**	0.66
C	6. Val. Set Metrics	-3.26	-0.04
All	9. Hyperparameter Bounds	6.61	0.07

Table 4: Multiple linear regression over items from the ACL Reproducibility Checklist as predictors for students' code setup time and runtime (% difference from research team). Most significant predictors for each assigned paper and overall are listed, along with their correlation coefficients β and adjusted R^2 values. Statistically significant ($p < 0.05$) coefficients in bold.

Paper	Top ACLRC Item, Setup Difficulty	β
A	10. Best Hyperparameters	**1.82**
B	11. Model Selection Strategy	**4.26**
C	5. Model Complexity Info	**-4.40**
All	15. Data Preprocessing Info	**0.65**

Table 5: Ordinal logistic regression over items from the ACL Reproducibility Checklist as predictors for students' reported code setup difficulty (1-5, 5 being most difficult). Most significant predictors for each assigned paper and overall are listed, along with their correlation coefficients β. Statistically significant ($p < 0.05$) coefficients in bold.

Relationship with difficulty. In Table 5, we similarly analyze how papers' reproducibility efforts affected students' reported code setup difficulty, which earlier showed significant variations with respect to skill level.[16] These results again indicate model hyperparameters (Paper A) and model selection strategy (Paper B) to be significantly associated with higher reported difficulty. Meanwhile, the reporting of model complexity was significantly associated with lower difficulty in Paper C, suggesting authors may have provided helpful information related to this. This may provide some more clues toward paper-specific variations in students' experience.

Open-ended feedback. Beyond reproducibility efforts from the ACLRC, we surveyed students directly for additional open-ended feedback on their experience and outlooks on the experiments. We asked students what aspects of their assigned pa-

---

[15]Items 2 and 17, which refer to the prerequisite steps of authors simply releasing their code and data, are omitted from analysis due to high variance inflation factor.

[16]See more about difficulty ratings for each paper in Appendix B.2.

Reproducibility Helper	Frequency
Clear Code Usage Documentation	56
Example Scripts and Commands	27
Easy-to-Read Code	15
Easy-to-Access External Resources	13
Sufficient Code Dependency Specification	12
Other	11

Table 6: Top 5 reported categories of features that helped students' reproduction of results. Less frequent responses aggregated in Other category.

Reproducibility Blocker	Frequency
Insufficient Code Dependency Specification	38
Difficult-to-Access External Resources	27
Unclear Code Usage Documentation	17
Pre-Existing Bugs in Code	16
Difficult-to-Read Code	11
Other	30

Table 7: Top 5 reported categories of features that blocked students' reproduction of results. Less frequent responses aggregated in Other category.

pers helped or blocked them in reproducing the results, as well as what should be added to the ACLRC to improve their experience. We categorized student responses and aggregated them in Tables 6, 7, and 8. Comments focused on several aspects, varying by their assigned paper and unique experience with reproducing its results. Nonetheless, common responses provide rich insights about what NLP beginners need most to get their hands on the latest research. These insights, primarily relating to engineering issues in using released code and data, are summarized in Section 5.

5 Discussion & Recommendations

Our study reveals a wealth of insights into enhancing the accessibility of NLP research to beginners.

Suggested ACLRC Addition	Frequency
Standards for Documentation Clarity	22
Full Specification of Code Dependencies	18
Demonstration of Code Usage	9
Provision of Support for Issues	8
Standards for Code Clarity	5
Other	23
Already Included	23

Table 8: Top 5 suggested categories of additions to the ACL Reproducibility Checklist (ACLRC). Less frequent suggestions and those already addressed in the ACLRC aggregated in Other and Already Included categories.

The most interesting insight is that deliberate reproducibility efforts by authors beyond simply writing a paper and releasing the code are more crucial to beginners' experience in reproducing results than their programming skills and paper comprehension. This finding behooves us researchers to diligently make these efforts, which would result in a win-win situation: people outside of the NLP research community, e.g., researchers from other disciplines and even the general public, can engage with our research more, which will extend the impact of our research.

Lastly, we share concrete, actionable recommendations on how to do this, framed around students' common feedback in Tables 6, 7, and 8. Where we find that current reproducibility guidelines for NLP research are insufficient, we make recommendations on how they may be strengthened to consider beginners' experiences.

Code dependencies. The most common complaint from students (reported for all papers) was the specification of code dependencies, e.g., the versions of Python and Python packages. On the ACLRC, this effort is only briefly mentioned in Item 2, a general item about open-sourcing the code which does not reference version numbers of dependencies. Consequently, including more details about dependencies, especially version numbers, was the second most common suggestion by students to add to the ACLRC. In contrast, the Responsible NLP Research checklist (RNLPRC) recently adopted at more ACL venues[17] emphasizes these efforts. Fortunately, NLP researchers can rely on various computing environment management tools, such as pip[18], conda[19], Poetry,[20] and Docker.[21] Simply utilizing such tools when sharing our work can make a meaningful difference for beginners.

Instructions for reproducing results. Just releasing source code is not enough for others to reproduce results; it needs to be accompanied by clear usage documentation with steps to reproduce the results. Documentation was the most appreciated effort by students, and also the third most common complaint, suggesting that it can make or break a beginner's experience in reproducing

---

[17] https://aclrollingreview.org/responsibleNLPresearch/
[18] https://pypi.org/project/pip/
[19] https://docs.conda.io/en/latest/
[20] https://python-poetry.org/
[21] https://www.docker.com/

results. Standards for code documentation were the most common suggested addition to the ACLRC by students. Good code documentation is a huge topic, and there are many resources available for this matter (Parnas, 2011; Aghajani et al., 2019; Hermann and Fehr, 2022). Furthermore, students' specific feedback can also provide inspiration here. For example, one common suggestion for documentation standards on the ACLRC was clearly documenting the correspondence between code and results in the paper to highlight how to reproduce those results as reported. Related to documentation, students' second most appreciated effort was providing examples for code usage, and the third most suggested addition to the ACLRC was a demonstration of code usage, whether it be an example script or command, an interactive notebook, or even a video. Neither the ACLRC or RNLPRC include specific recommendations for code usage instructions or documentation. As such, we recommend that clear documentation of code usage be considered as a criterion, and it may be worthwhile for the ACL community to propose some standards for future work to follow.

Availability of external resources. While all code was released on GitHub, large artifacts (e.g., datasets or pre-trained models) that are not suitable for `git` were hosted on other websites. Students reported difficulties when attempting to access them, such as broken links and dependency on third-party software. Conversely, students commonly rated ease of access to such resources helpful to reproducing results. While providing access to external resources is already suggested in the ACLRC, it is not explicitly mentioned in the RNLPRC, which may give an impression that this is not important, despite being essential to reproducibility. As such, we recommend that this issue be discussed in the RNLPRC. Further, we suggest that NLP researchers should take extra care by using a centralized system like `HuggingFace Datasets` (Lhoest et al., 2021),[22] or, at a minimum, periodically verifying that important resources are still accessible.

Code clarity and functionality. Students found code clarity, including informative code comments and variable names, neatness of code, and intuitive file structures, to be the third most helpful effort in papers' code bases. While the ACLRC and RNLPRC make no recommendations for this,

this was a common suggestion to add. Thankfully, there are widely agreed upon standards for code clarity (Martin, 2009; Wilson et al., 2014), and automated formatting tools like `black`[23] can make this easier. Further, many students were blocked by minor bugs in the code when reproducing results. Authors should take extra care to avoid them in order to enable others to engage with their work without frustration. One common student suggestion for the ACLRC was to provide support for bugs, whether interactively through forums like GitHub Issues[24] or proactively through an FAQ in their documentation. Another less common suggestion was to perform a sanity check reproduction of results on a clean copy of the code before open-sourcing and after substantial changes. Such an effort to mitigate even minor code bugs could make a substantial difference in the reproduction experience for beginners.

Addressing these key issues in NLP research practice[25] could greatly improve the experience of beginners when reproducing experimental results, and extend the accessibility of the latest research developments to those even beyond the research community. As NLP and AI research have recently attracted unprecedented global attention, we encourage the community to continue to dive deeper into the outlooks of beginners in future work. For example, given the recent paradigm shift in NLP from fine-tuning pre-trained language models to applying them directly to downstream tasks, there may also be a shift in user reproducibility roadblocks that will be important for the community to understand as it continues to strive for reproducible and accessible research. While some issues will remain important (e.g., code dependencies or data availability), other issues may become less crucial (e.g., hyperparameter search), while completely new issues may appear (e.g., choice of prompts used with a model). More broadly, there are also myriad opportunities to explore other topic areas and subject populations, the differences in reproducibility experiences between experts and beginners, and beginners' perception of state-of-the-art NLP systems and how they interact with them.

---

[22] https://huggingface.co/docs/datasets

[23] https://black.readthedocs.io/en/stable/

[24] https://github.com/features/issues

[25] While our study focused on NLP research, many of the resulting insights are also applicable to neighboring disciplines.

Acknowledgements

Limitations

Study scope. While our study only considers three papers, this is by design. As our goal is to study user experience, by fixing papers to be within a specific topic area and time requirement, and having people with different skill levels reproduce the same papers, this allows us to have sufficient samples to understand general behaviors. It also blocks other nuance factors (e.g., introduced by different papers) on beginners' experience. Each of our selected papers presented students with unique reproducibility barriers, and consequently resulted in a wealth of helpful insights. Furthermore, finding reproducible NLP papers that satisfied our constraints (as laid out in Section 3.1.2) was surprisingly difficult, with only 3 out of 24 considered papers found to be reproducible within our constraints. Nevertheless, this study is still on the small scale. Engaging a larger community in a large scale study may provide additional insight. Related to this, our study only includes a population of mostly graduate students at our university. Considering beginners from different educational backgrounds or regions could reveal more comprehensive insights, and we greatly encourage future efforts at a community level toward better understanding the needs of NLP beginners.

GPU runtime calculation. It is also worth noting that it is difficult to consistently calculate runtime (as introduced in Section 3.2.1) of code on GPU hardware, as fluctuations may occur due to a number of factors, including the specific GPU hardware allocated to a student,[27] driver versions, and file systems experiments were run with. To minimize the impact of such issues, we chose to reproduce experiments that used small models and had shorter expected runtimes. Given that we observed runtimes up to several times larger than expert runtimes, we thus expect that trial and error in setting up experiments accounted for most fluctuation in observed runtimes.

Ethics Statement

This work was based on a study conducted as part of a graduate-level course including survey responses and computational resource usage statistics. Our institution's Institutional Review Board (IRB) approved this human subjects research before the start of the study.[28] Subjects completed an informed FERPA-compliant consent form to opt into the study and were not compensated, since the collected data was part of a regular homework assignment. As the research team for this work was also the instructional team of the course, one key ethical issue we aimed to mitigate was subjects feeling pressured to consent to this research in hopes it may benefit their grades. As such, we designated one member of the research team who was unable to view or modify student grades. Only this team member had access to informed consent responses from the students, and then linked and de-identified data before sharing it with the rest of the team. De-identification of data included classifying all free-text responses into a number of class labels so that students could not be recognized from their responses. Students were made aware that their participation was entirely optional, and could not possibly impact their grade due to this careful arrangement. Further, to ensure that students were assigned a comparable amount of work, we carefully selected papers with results that could be reproduced by the research staff in a comparable amount of time (i.e., 2 hours).[29]

The results of this study could have a positive impact on the NLP research community, as it reveals insights that may be helpful for NLP researchers to better enable beginners to get their hands on research artifacts and reproduce their results. If ap-

---

[26] https://arc.umich.edu/

[27] While experts used NVIDIA Tesla V100 GPUs with up to 16GB memory to reproduce results, NVIDIA A40 GPUs with up to 48GB memory are also available within the cluster.

[28] Human subjects research approved by University of Michigan Health Sciences and Behavioral Sciences Institutional Review Board (IRB-HSBS), eResearch ID HUM00224383.

[29] As mentioned in Appendix C, one considered paper was discarded due to taking a significantly shorter amount of time than the others.

plied in future open-sourcing of research artifacts, such insights could expand the accessibility of our work to a broader audience. As NLP systems are becoming more ubiquitous in society and attracting attention beyond our research community, this effort could result in the inclusion of more voices in discussions around them and their future development, which is essential for democratization.

References

Emad Aghajani, Csaba Nagy, Olga Lucero Vega-Márquez, Mario Linares-Vásquez, Laura Moreno, Gabriele Bavota, and Michele Lanza. 2019. Software documentation issues unveiled. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 1199–1210.

João António Rodrigues, Ruben Branco, João Silva, and António Branco. 2020. Reproduction and revival of the argument reasoning comprehension task. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5055–5064, Marseille, France. European Language Resources Association.

Sylwester Arabas, Michael R Bareford, Lakshitha R de Silva, Ian P Gent, Benjamin M Gorman, Masih Hajiarabderkani, Tristan Henderson, Luke Hutton, Alexander Konovalov, Lars Kotthoff, et al. 2014. Case studies and challenges in reproducibility in the computational sciences. *arXiv preprint arXiv:1408.2123*.

Jaime Arguello, Matt Crane, Fernando Diaz, Jimmy Lin, and Andrew Trotman. 2016. Report on the SIGIR 2015 workshop on reproducibility, inexplicability, and generalizability of results (RIGOR). *SIGIR Forum*, 49(2):107–116.

Monya Baker. 2016. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604).

Anya Belz. 2022. A Metrological Perspective on Reproducibility in NLP*. *Computational Linguistics*, 48(4):1125–1135.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2021. A systematic review of reproducibility research in natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 381–393, Online. Association for Computational Linguistics.

António Branco, Nicoletta Calzolari, Piek Vossen, Gertjan Van Noord, Dieter van Uytvanck, João Silva, Luís Gomes, André Moreira, and Willem Elbers. 2020. A shared task of a new, collaborative type to foster reproducibility: A first exercise in the area of language science and technology with REPROLANG2020. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5539–5545, Marseille, France. European Language Resources Association.

K. Bretonnel Cohen, Jingbo Xia, Pierre Zweigenbaum, Tiffany Callahan, Orin Hargraves, Foster Goss, Nancy Ide, Aurélie Névéol, Cyril Grouin, and Lawrence E. Hunter. 2018. Three dimensions of reproducibility in natural language processing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Matt Crane. 2018. Questionable answers in question answering research: Reproducibility and variability of published results. *Transactions of the Association for Computational Linguistics*, 6:241–252.

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)*, pages 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Lucia Donatelli, Theresa Schmidt, Debanjali Biswas, Arne Köhn, Fangzhou Zhai, and Alexander Koller. 2021. Aligning actions across recipe graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6930–6942, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Janet Feigenspan, Christian Kästner, Jörg Liebig, Sven Apel, and Stefan Hanenberg. 2012. Measuring programming experience. In *2012 20th IEEE International Conference on Program Comprehension (ICPC)*, pages 73–82. IEEE.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Sibylle Hermann and Jörg Fehr. 2022. Documenting research software in engineering science. *Scientific Reports*, 12(1):1–11.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas

Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robert C Martin. 2009. *Clean code: a handbook of agile software craftsmanship*. Pearson Education.

David Lorge Parnas. 2011. *Precise Documentation: The Key to Better Software*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Joelle Pineau. 2020. The machine learning reproducibility checklist.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. 2021. Improving reproducibility in machine learning research: a report from the NeurIPS 2019 reproducibility program. *Journal of Machine Learning Research*, 22.

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*, pages 4821–4833, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicolas P Rougier, Konrad Hinsen, Frédéric Alexandre, Thomas Arildsen, Lorena A Barba, Fabien CY Benureau, C Titus Brown, Pierre De Buyl, Ozan Caglayan, Andrew P Davison, et al. 2017. Sustainable computational science: the rescience initiative. *PeerJ Computer Science*, 3:e142.

Kristin Yvonne Rozier and Eric WD Rozier. 2014. Reproducibility, correctness, and buildability: The three principles for ethical public dissemination of computer science and engineering research. In *2014 IEEE International Symposium on Ethics in Science, Technology and Engineering*, pages 1–13. IEEE.

Ricky J. Sethi and Yolanda Gil. 2016. Reproducibility in computer vision: Towards open publication of image analysis experiments as semantic workflows. In *2016 IEEE 12th International Conference on e-Science (e-Science)*, pages 343–348.

R. Tatman, J. VanderPlas, and S. Dane. 2018. A practical taxonomy of reproducibility for machine learning research. In *Reproducibility in Machine Learning Workshop at ICML 2018*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Martijn Wieling, Josine Rawee, and Gertjan van Noord. 2018. Reproducibility in Computational Linguistics: Are We Willing to Share? *Computational Linguistics*, 44(4):641–649.

Greg Wilson, Dhavide A Aruliah, C Titus Brown, Neil P Chue Hong, Matt Davis, Richard T Guy, Steven HD Haddock, Kathryn D Huff, Ian M Mitchell, Mark D Plumbley, et al. 2014. Best practices for scientific computing. *PLoS biology*, 12(1):e1001745.

Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2021. Temporal reasoning on implicit events from distant supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1361–1371, Online. Association for Computational Linguistics.

A ACL Reproducibility Checklist

The full ACL Reproducibility Checklist is provided below.

- For all reported experimental results:

  1. A clear description of the mathematical setting, algorithm, and/or model;
  2. A link to (anonymized, for submission) downloadable source code, with specification of all dependencies, including external libraries;
  3. A description of the computing infrastructure used;
  4. The average runtime for each model or algorithm, or estimated energy cost;
  5. The number of parameters in each model;
  6. Corresponding validation performance for each reported test result;
  7. A clear definition of the specific evaluation measure or statistics used to report results.

- For all results involving multiple experiments:

  8. The exact number of training and evaluation runs;
  9. The bounds for each hyperparameter;
  10. The hyperparameter configurations for best-performing models;
  11. The method of choosing hyperparameter values (e.g., manual tuning, uniform sampling, etc.) and the criterion used to select among them (e.g., accuracy);
  12. Summary statistics of the results (e.g., mean, variance, error bars, etc.).

- For all datasets used:

  13. Relevant statistics such as number of examples and label distributions;
  14. Details of train/validation/test splits;
  15. An explanation of any data that were excluded, and all pre-processing steps;
  16. For natural language data, the name of the language(s);
  17. A link to a downloadable version of the dataset or simulation environment;
  18. For new data collected, a complete description of the data collection process, such as ownership/licensing, informed consent, instructions to annotators and methods for quality control.
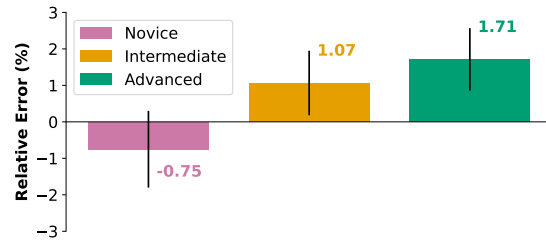

Figure 5: Average reported relative error from paper results, by students' technical skill level.
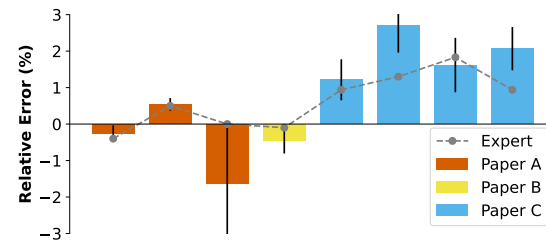

Figure 6: Relative error of students' reported results from those reported in the paper for each evaluation setting, compared to our expert result (dotted line).

B Supplementary Results

Here, we include some extra results that, while significant or informative, were less relevant to the message of our paper.

B.1 Reproduced Accuracy

For each paper, we asked students to re-train one NLP system and report the accuracy in the same settings and on the same data partitions as reported in the paper. Figure 5 averages the relative error of students' submitted results by experience group. As shown, on average, student results do not vary significantly by experience level. Further, on average, student results come close to those reported in the paper, and for the most part, do not differ significantly from our reproduced results.

In Figures 5 and 6, we compare the relative error between students' reported results for each skill level, and for each setting of the reproduced system and the results published in their corresponding papers. In both cases, the student-obtained results are fairly aligned to the reported results in the paper (as well as our own reproduced results), with standard errors within 3% accuracy.

B.2 Reproducibility Difficulty by Paper

Figure 7 summarizes student ratings for how difficult each step of the exercise was. Code setup, data preprocessing, and system training unsurprisingly had the lowest ratings, and mean ratings did not
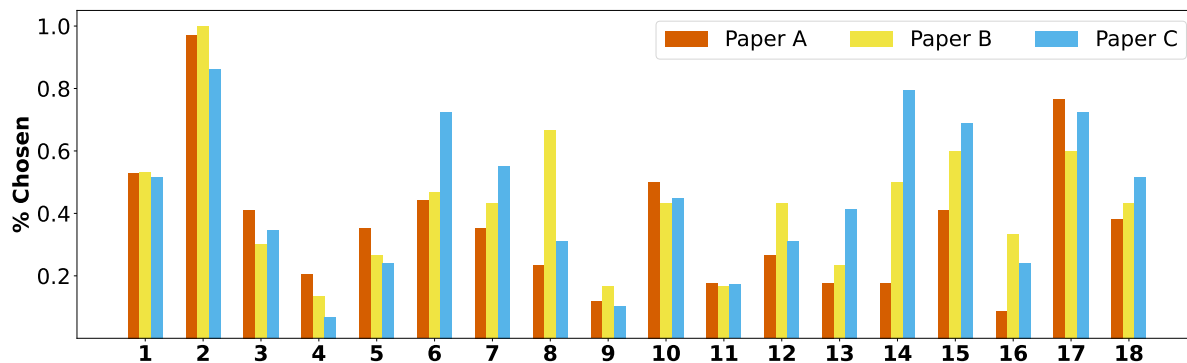
Figure 8: Percentage of responses in which each ACL Reproducibility Checklist item (indexed by the ordering of items) was selected as helpful by students in the study, by paper. See Appendix A for full text of checklist.
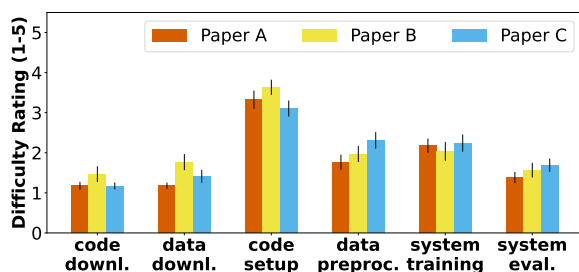
vary much across papers.

Figure 7: Mean reproducibility difficulty rating (1-5, higher being most difficult) for each step of experiments.

B.3 ACL Reproducibility Checklist Survey

After completing their assigned work, students indicated which items on the ACL Reproducibility Checklist were most helpful in reproducing the results. In Figure 8, we show the percentage of times each item was selected, aggregated by paper. The differences between items in the graph may give further insights on which parts of the checklist are most helpful to NLP beginners, and where our studied papers differed in terms of what was provided in open-sourced materials.

C Paper Selection

To find experiments to reproduce for the study, we considered papers from ACL conferences, which are top venues for NLP research. We specifically collected papers from the Annual Meeting of the ACL, the Conference on Empirical Methods in Natural Language Processing (EMNLP), and the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) from the years 2018 through 2022. Over these conferences, we arbitrarily selected 24 long papers from two topic areas: (A) *Semantics: Sentence-level Semantics*

and Textual Inference,[30] and (B) *Language Generation.* All selected papers had publicly available code, models (where applicable), and data with licenses permitting academic research use.

For each selected paper, we attempted to reproduce the results of an experiment ourselves. If a paper presented several experiments, e.g., for different model variations, we chose the best-performing instance based on the evaluation metrics applied. If the best-performing instance could not be reproduced within a feasible amount of time or compute resources, we chose the next best that fit within our computing restrictions. For our expert reproduction of results, we limited the effort to 2 hours of setup time and 4 hours of total code runtime (for successful runs of the code) on a single GPU[31] for any model training or evaluation, while our computing restrictions were dictated by the available hardware and software within the environment used to reproduce results.[32]

Out of these 24 papers, the results of only 4 papers could be successfully reproduced within the above constraints, all of which belonged to Area A. As this research was conducted as part of a homework assignment, we discarded one of these 4 papers that took a significantly shorter time to reproduce than the others. Common reasons that the experiment failed included long model training time requirements, incompatibility of code bases with our computing platform, incompleteness of documentation, and discrepancy between repro-

---

[30]Prior to EMNLP 2020, this area was referred to as *Semantics: Textual Inference and Other Areas of Semantics*, but was merged with another area.

[31]Setup time and runtime defined in Section 3.2.1.

[32]For reproducing all results, both experts and students used an onsite computing cluster that offers access to NVIDIA Tesla V100 GPUs with up to 16GB memory, and NVIDIA A40 GPUs with up to 48GB memory.

duced results and those reported in papers.

The specific papers we selected for the study are listed in Table 1, and the distribution of paper assignments across skill levels is listed in Table 2. In Appendix E, we describe the specific experiments that we reproduced, and thus assigned for students to reproduce.

D Pre-Survey Questions

As discussed in Section 3.1.1 we conducted a brief survey on students' background at the beginning of the study. As mentioned in Section 3.2.2, the results of this survey were used in characterizing students' skill levels. Pre-survey questions relevant to the study are listed below.

1. Before taking this course, how many years of experience did you have specifically with Python?

2. Before taking this course, how many years of experience did you have specifically with PyTorch?

3. To the best of your memory, how difficult was it for you to complete these past homework problems?

   Included pointers to specific homework problems on implementing LSTMs and transformers, and asked about them individually.

   - *very difficult*
   - *slightly difficult*
   - *neutral*
   - *slightly easy*
   - *very easy*

E Homework Assignment Content

We include some content from the homework assignment associated with this study which was used to prepare students for the work and prime them to report their results in the post-survey. Among course-related administrative details (e.g., how to access surveys, grading guidelines, etc.), we gave them important information on which experiment to reproduce from their assigned papers, what steps they should take when reproducing the results, and how to track their time spent on the experiment.

Assigned experiments. For each paper, the specific experiments we reproduced (and instructed students to reproduce) were as follows.

For Paper A, we instructed students to reproduce the following experiment from Zhou et al. (2021): *Fine-tune* PTNTIME *model to the uniform-prior TRACIE dataset, then evaluate it on the testing set (*PTNTIME *result with 76.6% "All" accuracy in Table 2 from the paper). Report the Start, End, and All accuracies on the testing set.*

For Paper B, we instructed students to reproduce the following experiment from Donatelli et al. (2021): *Train the base alignment model with* BERT *embeddings on the pre-tagged Action Alignment Corpus, and evaluate on the testing set ("Our Alignment Model (base)" result in Table 4). Report the combined accuracy on the testing set after cross-validation.*

For Paper C, we instructed students to reproduce the following experiment from Gupta et al. (2020): *Train* ROBERTA-*base on the InfoTabs dataset with* TABFACT *structured premise representation, and evaluate it on the development set and all three testing sets (*ROBERTAB/TABFACT *result with 68.06% Dev accuracy in Table 7 from the paper). Report the accuracy on the development set and all three testing sets.*

Reading the paper. *The first step for this assignment is to read the paper. This will make clear the experiments and results the paper has contributed, so that when you begin to reproduce the results, you will already know what to expect.*

Reproducing the results. *Next, you should open the code base for your paper and follow the authors' instructions for re- producing the results (only those experiments that we specify). Their documentation will usually include some combination of the following steps:*

- *Cloning a GitHub repo to your local environment (make sure you have an account)*
- *Installing Python dependencies*
- *Downloading datasets and pre-trained models*
- *Training and validating models*
- *Testing models*

Different papers will give varying degrees of guidance for these steps. Documentation for a paper's code release may have typos or missing information. You should use your understanding of the paper, Python, and PyTorch to fill in any gaps.

Tracking your time. *Please carefully track the time you spend on the homework assignment. Categorize your time into the following activities:*

1. *Closely **reading** the paper*
2. *Setting up the **code base** to successfully run experiments, e.g., downloading the code and data, preprocessing data, and installing dependencies*
3. ***Training** models (i.e., waiting for training code to run)*
4. ***Evaluating** models (i.e., waiting for evaluation code to run)*
5. ***Documenting** results and completing the post-survey*

You will be asked for this information in the post-survey to help us understand the difficulties that you ran into. Please report your time honestly; your grade will not depend on how much time it took to complete the assignment.

F Post-Survey Questions

As discussed in Section 3.1.4, students were asked to complete a survey after reproducing their assigned results.[33] Post-survey questions relevant to the study are listed below.

1. *A series of paper-specific reading comprehension questions, listed in Appendix G.*

2. Were you able to reproduce the results of the paper?

3. Please report the performance of your trained model under each of the evaluation settings.

   See Appendix E for specific results reported for each paper.

4. How much time did it take you to set up the code base for model training? Count from the time the code base was downloaded until model training was successfully running, including data preprocessing. Don't count time spent actually running training code. *(in hours and minutes)*

5. How much time did it take you to set up the code base for model evaluation? Count from the time the code base was downloaded until model evaluation was successfully running, including data preprocessing. Don't count time spent actually running evaluation code. *(in hours and minutes)*

---

[33]Students were primed for some of these questions through assignment documents we distributed to introduce the expected work and students' assigned papers and experiments. Relevant content is included in Appendix E.

6. Please rate the level of difficulty you encountered in each step of the experiment. *(1: very difficult - 5: very easy)*

   • *Downloading the code base.*
   • *Downloading required data.*
   • *Setting up the code base and its dependencies.*
   • *Preprocessing the data.*
   • *Training models.*
   • *Evaluating models.*

7. Did the authors provide anything in the code release that helped make the results easier for you to reproduce?

8. What could the authors have provided or done better for you to reproduce the results faster or with less frustration?

9. Which of the following items from the ACL Reproducibility Checklist were especially helpful for reproducing the paper's results? Please check all that apply.

   Checklist items listed in Appendix A.

10. Is there anything you would add to this checklist that you wish authors would provide to help you reproduce results faster or with less frustration? Suggest up to 5.

G Comprehension Questions

Here, we list the specific comprehension questions used in the post-survey for each paper (correct answers in bold).

G.1 Paper A Questions

1. **Motivation:** Which of the following is not a motivation of this work?

   (a) *Humans can recognize temporal relationships between events both explicitly mentioned and implied (but not explicitly mentioned) in language.*

   (b) *Past work in temporal reasoning has focused only on explicitly mentioned events.*

   (c) ***At the time of writing, there were no benchmark datasets evaluating temporal reasoning for NLP.***

   (d) *Constructing a latent timeline of events is essential for NLP systems to understand stories.*

2. **Problem Definition:** What task is being studied in this paper?

   (a) ***Textual entailment with a focus on ordering of events.***

   (b) *Question answering with a focus on temporal reasoning.*

   (c) *Story generation with a coherent timeline of events.*

   (d) *Semantic role labeling for implicit events.*

3. **Approaches:** What are the inputs to the PT-NTIME system?

   (a) *A short story, a timeline of explicit and implicit events, and a proposed ordering of events.*

   (b) *A story written in text, a hypothesis written in text, and an inference label.*

   (c) *A story written in text, with BIO tags describing the spans of text where a specific event occurs.*

   (d) ***A story written in text, and a hypothesis about the temporal ordering of events in the story.***

4. **Approaches:** What are the outputs of the PT-NTIME system?

   (a) *Tags describing the span of text the target event occurs.*

   (b) ***An inference label of entailment or contradiction.***

   (c) *An inference label of entailment, neutral, or contradiction.*

   (d) *A timeline of explicit and implicit events in the text.*

5. **Implementation:** Which of the following files from the paper's code base is responsible for fine-tuning PTNTIME to TRACIE?

   (a) `tracie/code/models/symtime/train_t5.py`

   (b) `tracie/code/models/ptntime/evaluator.py`

   (c) **`tracie/code/models/ptntime/train_t5.py`**

   (d) `tracie/code/models/ptntime/train_ptntime.py`

6. **Results:** What is the meaning of the values in the "Story" column in Table 1 of the paper?

   (a) *The average percentage of hypotheses which were correctly classified for each story.*

   (b) *The percentage of stories which were correctly classified before a hypothesis was introduced.*

   (c) *The percentage of stories for which the system made a correct prediction on any hypothesis.*

   (d) ***The percentage of stories for which the system made correct predictions on all hypotheses.***

7. **Conclusion:** Which of the following CANNOT be concluded based on the results in this paper?

   (a) *Symbolic temporal reasoning can be used to improve language models' temporal reasoning.*

   (b) *Distant supervision on event durations can be used to improve language models' temporal reasoning.*

   (c) *The uniform-prior setting of TRACIE is harder for baseline systems to solve than the full dataset.*

   (d) ***When used in a zero-shot setting, the proposed models consistently outperform baselines from prior work fine-tuned on the task.***

G.2 Paper B Questions

1. **Motivation:** Which of the following is NOT a motivation of this work?

   (a) *A key challenge with interpreting cooking recipes is that for any dish, different recipes may omit or emphasize different steps.*

   (b) ***Aligning recipes based only on actions ignores rich information about the structure of recipes and relationships between sentences.***

   (c) *Aligning multiple recipes for a single dish would give a recipe understanding system more complete information about making that dish.*

   (d) *Past work has not looked into aligning multiple recipes at the action level.*

2. **Problem Definition:** What task is being studied in this paper?

   (a) ***Alignment of actions in multiple recipes for the same dish.***

(b) *Alignment of sentences in multiple recipes for the same dish.*

(c) *Alignment of actions in recipes for similar (but not identical) dishes.*

(d) *Alignment of sentences across recipes for similar (but not identical) dishes.*

3. **Approaches:** What are the inputs to the base alignment model?

   (a) *One sentence describing one or more actions in a recipe.*

   (b) *A span of steps in a recipe, each of which is a sentence.*

   (c) ***Two recipes for a single dish, and a source action selected from one of the recipes.***

   (d) *Two recipes for two different dishes, and a source action selected from one of the recipes.*

4. **Approaches:** What are the outputs of the base alignment model?

   (a) *All actions from the target recipe such that their confidence scores for aligning to the given source action exceed a threshold value.*

   (b) *A single action from the target recipe which best aligns to the given source action.*

   (c) ***Either one action from the target recipe which best aligns to the given source action, or no matching actions.***

   (d) *The top five actions from the target recipe which best align to the given source action.*

5. **Implementation:** In which file are confidence scores from the alignment model calculated?

   (a) `ara/Alignment_Model/main.py`

   (b) `ara/Alignment_Model/utils.py`

   (c) `ara/Alignment_Model/
   training_testing.py`

   (d) **`ara/Alignment_Model/model.py`**

6. **Results:** How are the alignment model results from Table 4 in the paper calculated?

   (a) *The model is trained on the training set, validated on a validation set consisting of recipes for the same dishes as the training set, then tested on a set of recipes for dishes not seen in training or validation. The accuracy on this testing set is reported in Table 4.*

(b) ***Ten instances of the alignment model are trained using cross validation, where each fold holds out the recipes from one of the ten dishes as validation data, and from another dish as testing data. The testing results on each of the ten dishes are combined from the ten model instances.***

(c) *The model is trained on the training set, validated on a validation set consisting of recipes for the same dishes as the training set, then tested on a set of held-out recipes for those same dishes. The accuracy on this testing set is reported in Table 4.*

(d) *Ten instances of the alignment model are trained using cross validation, where each fold holds out one of the ten dishes as testing data, and a validation set of recipes is randomly sampled from the training data. The testing results on each of the ten dishes are combined from the ten model instances.*

7. **Conclusion:** Which of the following CANNOT be concluded based on the results in this paper?

   (a) ***The alignment models struggle to generalize, as they perform better on recipes for dishes seen in training than those not seen in training.***

   (b) *Simply aligning recipe actions based on their sequential order is not a viable baseline for the task, but using cosine similarity works better.*

   (c) *Incorporating graphical information about a recipe improves the alignment model's performance on aligning recipe actions.*

   (d) *None of the proposed systems achieve human performance, demonstrating the difficulty of the recipe alignment problem.*

G.3 Paper C Questions

1. **Motivation:** Which of the following is NOT a motivation of this work?

   (a) *Understanding tables requires reasoning over multiple fragments of text in different cells that may not otherwise seem related.*

   (b) ***Tables are uniquely challenging to understand because they convey explicit information that unstructured text does not.***

   (c) *Transformer-based language models have exceeded human performance on a variety of natural language understanding tasks.*

(d) *Semi-structured text can convey unstated information that state-of-the-art language models may fail to recognize.*

2. **Problem Definition:** What task is being studied in this paper?

   (a) *Question answering based on Wikipedia info-boxes.*

   (b) *Relation extraction for cells in Wikipedia info-boxes.*

   (c) *Table information summarization.*

   (d) ***Textual entailment based on a semi-structured context.***

3. **Approaches:** What are the inputs to the ROBERTA baseline model with TabFact structured premise representation?

   (a) *A table converted to a paragraph, and a proposed fact about the table.*

   (b) ***A table converted to a set of key-value pairs, and a proposed fact about the table.***

   (c) *A proposed fact about the table, and the most similar sentence to it from the table.*

   (d) *A proposed fact about the table, and the most similar three sentences to it from the table.*

4. **Approaches:** What are the outputs of the ROBERTA baseline model with TabFact structured premise representation?

   (a) *A one-sentence summary of the table in text.*

   (b) *An inference label of entailment or contradiction.*

   (c) ***An inference label of entailment, neutral, or contradiction.***

   (d) *Tags describing the span of cells that answer the question about the table.*

5. **Implementation:** In which file are tables converted to TabFact structured premises?

   (a) **infotabs-code/scripts/preprocess/ json_to_struct.py**

   (b) infotabs-code/scripts/preprocess/ json_to_wmd.py

   (c) infotabs-code/scripts/roberta/ preprocess_roberta.py

   (d) infotabs-code/scripts/roberta/ json_to_struct.py

6. **Results:** What is the difference between the "α_2" and "α_3" columns in Table 7 of the paper?

   (a) *The α_2 test set includes tables from a different domain than the training set, while the α_3 test set includes hypothesis sentences that have been adversarially edited by human annotators.*

   (b) *The α_2 test set includes hypothesis sentences from a different domain than the training set, while the α_3 test set includes tables that have been adversarially edited by human annotators.*

   (c) ***Models that overfit to superficial lexical cues will struggle with the α_2 test set, while models that overfit to domain-specific statistical cues will struggle with the α_3 test set.***

   (d) *Models that overfit to domain-specific statistical cues will struggle with the α_2 test set, while models that overfit to superficial lexical cues will struggle with the α_3 test set.*

7. **Conclusion:** Which of the following CANNOT be concluded based on the results in this paper?

   (a) *Pre-trained state-of-the-art natural language inference systems do not perform well when applied directly to tasks requiring reasoning over tables.*

   (b) ***A support vector machine performs better than transformer-based language models on InfoTabs when representing tables as paragraphs.***

   (c) *Encoding a table in structured language rather than an unstructured paragraph helps improve performance of language models on InfoTabs.*

   (d) *The proposed systems for InfoTabs tend to struggle most with cross-domain generalization.*

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations are discussed after Section 5.

☑ A2. Did you discuss any potential risks of your work?
The Ethics Statement after Section 5 discusses how we mitigated possible risks to human subjects.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3 discusses how papers for the study are selected and their results reproduced. Also, as stipulated by our IRB and subject consent agreement, we will not be releasing the human subjects data collected in this study.

☑ B1. Did you cite the creators of artifacts you used?
Section 3.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix A.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
In Section 3, we introduce the study design - artifacts were used for a reproducibility study rather than for further empirical research.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The Ethics Statement after Section 5 discusses data de-identification.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. For this information, we refer readers to the original papers that created the artifacts we used, cited in Table 1.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. For this information, we refer readers to the original papers that created the artifacts we used, cited in Table 1.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☒ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
No response.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
No response.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
No response.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

See Section 3 and Ethics Statement after Section 5 for information about how human subjects participated in the study.

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
The Appendix includes survey questions the students answered, and some other relevant content we used to convey information about their assigned work.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Ethics Statement after Section 5 provides information about how participation in the study was optional, and how we mitigated subjects' pressure to participate.

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Ethics Statement after Section 5.

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
IRB approval information listed in Ethics Statement after Section 5.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
We reported that study subjects were students in a graduate-level course at our institution, as well as some aggregate information about subjects' educational and technical background. Other demographic details were not collected as part of this study.