Exploring Variation of Results from Different Experimental Conditions

Maja Popović,[1] **Mohammad Arvan,**[2] **Natalie Parde,**[2] **Anya Belz**[1]
[1]ADAPT Centre, School of Computing, DCU, Ireland
name.surname@adaptcentre.ie
[2]Department of Computer Science, University of Illinois Chicago
{marvan3,parde}@uic.edu

Abstract

It might reasonably be expected that running multiple experiments for the same task using the same data and model would yield very similar results. Recent research has, however, shown this not to be the case for many NLP experiments. In this paper, we report extensive coordinated work by two NLP groups to run the training and testing pipeline for three neural text simplification models under varying experimental conditions, including different random seeds, run-time environments, and dependency versions, yielding a large number of results for each of the three models using the same data and train/dev/test set splits. From one perspective, these results can be interpreted as shedding light on the reproducibility of evaluation results for the three NTS models, and we present an in-depth analysis of the variation observed for different combinations of experimental conditions. From another perspective, the results raise the question of whether the averaged score should be considered the 'true' result for each model.

1 Introduction

Recently there has been a promising surge of interest in reproducibility of NLP models, supported by challenges (Pineau et al., 2021), shared tasks (Belz et al., 2020), conference tracks (Carpuat et al., 2022), and even the *Reality Check* theme at this conference. The outcome of this surge in interest has been a flurry of reproducibility studies and related investigations (Belz et al., 2022a; Arvan et al., 2022a; Chen et al., 2022b). However, the collective findings from these efforts have been alarming.

With interest in reproducibility growing, the evidence is mounting that scores are substantially affected by changes not only to arbitrary factors like random seed and different data splits, but also by incidental factors such as the type of GPU on which an experiment is run, and the run-time environment. In many cases, near-identical scores can be guaranteed only when an experiment is re-run in fully containerised form. In effect, this means that even perfect sharing of information (once regarded as the answer to all our reproducibility problems[1] (Sonnenburg et al., 2007)) cannot guarantee identical results in all cases.

All this raises questions about reporting, experimental design and the informativeness of scores regarding the relative merits of different methods. Underlying these is the question of where the boundary lies – seemingly between the two extremes. On the one hand, exploration of methodological variations and reporting of separate scores is part and parcel of method development. On the other hand, arbitrary and incidental factors such as random seed are not part of method development, because they do not generalise to future applications of the same method. For the former, clearly, comparing and reporting different scores is important; for the latter, how to interpret, address or report variation in scores is an open question.

In this paper, we tackle this question by conducting a systematic and comprehensive investigation coordinated across two NLP groups to study the variation of the results across three neural text simplification (NTS) models under many different experimental conditions. We experiment with different random seeds, run-time environments, and dependency versions to ensure broad coverage of our study. We observe that reporting average score and its coefficient of variation is a more reliable standard than reporting the maximum value, and we urge researchers to record all methodological conditions, control incidental ones, and abstract away arbitrary factors to promote the reproducibility of their scientific contributions.

[1]"Reproducibility would be quite easy to achieve in machine learning simply by sharing the full code used for experiments" (Sonnenburg et al., 2007).

Findings of the Association for Computational Linguistics: ACL 2023, pages 2746–2757
July 9-14, 2023 ©2023 Association for Computational Linguistics

2 Task and Experimental Set-up

Our starting point for this exploration is the first neural text simplification system reported by Nisioi et al. (2017). This work was selected because it is suitable for our purposes: the authors provided a repository[2] which contains comprehensive information about the original work and the resources, thus facilitating repeat runs of their experiments and exploration of variation on their experimental conditions, which is not often the case for NLP papers. Moreover, the work has been reproduced before (Cooper and Shardlow, 2020; Popović and Belz, 2021; Popović et al., 2022; Belz et al., 2022a; Arvan et al., 2022b) as part of the REPROLANG 2020 (Branco et al., 2020) and ReproGen 2021/2022 (Belz et al., 2021, 2022b) shared tasks, which represents another reference point to choose it.

In the following subsections we describe the four different systems (§2.2), the single data set/split and four text processing variants (§2.3), and the two evaluation methods (§2.4) which were included in our exploration, either because they were part of the original study or because we added them. §2.5 provides an overview of the incidental and arbitrary variation arising in our different runs which we also analysed.

2.1 Task Background

Briefly, text simplification aims to transform a specified text into a simpler form while retaining the same meaning. This is potentially useful for a broad range of real-world applications, because it makes the text readable and understandable for wider audiences and also easier to process by automatic NLP tools. The notion of simplicity itself may be tied to a variety of factors ranging from lexical complexity to content coverage or sentence/document structure. Automatic text simiplification (ATS) can be rule-based or data-based. Many data-based techniques approach the task of simplifying text by adopting methods from machine translation (MT), which is also the case for our experiments. Our work does not seek to develop innovations in ATS specifically, but rather to use ATS models as a convenient case study for studying variation of results. Nonetheless, we provide this background to facilitate fuller understanding of the problem scope and goals of the reproduced systems.

[2] https://github.com/senisioi/NeuralTextSimplification

2.2 Systems

Nisioi et al. (2017)'s original work is one of the first which explored neural networks for ATS (neural ATS, or NTS). They used Long Short-Term Memory (LSTM) recurrent neural networks with attention in an encoder-decoder architecture. Two models were trained: one standard neural MT model (which we call LSTM), and one (LSTM-w2v) using external pre-trained word2vec word representations (Mikolov et al., 2013). All their experiments were carried out using the openNMT tool[3] (Klein et al., 2017). The used version is the initial version based on LuaTorch,[4] released in December 2016.

The authors provided information about all necessary external libraries and specific Python and Lua dependencies, and also released the two models they trained (LSMT and LSTM-w2v). It is worth noting that the source code uses Python 2.7 and Torch. The Python environment uses older versions of openNMT, NLTK, and gensim. This version of openNMT is no longer maintained and most of the libraries and dependencies have become obsolete, and it is therefore advised not to use this version anymore but to switch to one of the two newer ones (openNMT-py based on PyTorch or openNMT-tf based on TensorFlow). Therefore, it has become extremely challenging to recreate the same environment to regenerate and retrain the models using the released source code.

Other than variation in the libraries and environments, we conduct a random search for the LSTM models using the original repository. In this scenario, all the hyper-parameters are kept the same except the random seed. Knowing that the random seed affects the weight initialisation, the data order used in training, and the sampling used in the generation, we suspected that we might observe a wide range of results.

Given that LSTM models generally have been superseded by transformer models (Vaswani et al., 2017), we additionally trained a transformer model on the data provided by the authors, using another publicly available tool, Sockeye.[5] We used two versions of the tool: the first version, based on MXNet (Hieber et al., 2018), and the newest (third) version based on PyTorch (Hieber et al., 2022). We treat these two versions as two different systems using the same model type. Thus to summarise,

[3] https://opennmt.net/

[4] https://github.com/OpenNMT/OpenNMT

[5] https://awslabs.github.io/sockeye/index.html

our systems are:

- **LSTM/OpenNMT:** Nisioi et al. (2017)'s LSTM neural MT model implemented as the first version of the OpenNMT tool.
- **LSTM-w2v/OpenNMT:** Nisioi et al. (2017)'s LSTM neural MT model, using external pre-trained word2vec representations implemented as the first version of the OpenNMT tool.
- **Transformer/Sockeye v1 (MXNet):** Our updated version of the NTS model, using a transformer model instead of an LSTM, implemented as the first version of the Sockeye tool based on MXNet.
- **Transformer/Sockeye v3 (PyTorch):** Our updated version of the NTS model, using a transformer model instead of an LSTM implemented as the newest (third) version of the Sockeye tool based on PyTorch.

We report results achieved under numerous conditions for each of these systems, ensuring broad coverage and supporting the robustness of the investigation.

2.3 Data Set and Text Processing

Nisioi et al.'s (2017) repository contains the pre-processed data set, but not the original data nor the pre-processing scripts. Their data set was a popular corpus of parallel English Wikipedia and Simple English Wikipedia (EW-SEW) articles (Hwang et al., 2015), and we used the same data for our experiments. The corpus statistics for the parallel data in both the training and tests sets are presented in Table 1. We report the number of sentences and words and the overall vocabulary size for each partition (original/simplified × train/test) of the data.

In the original paper, it is reported that Named Entities were treated separately: they were first identified, then replaced by an 'unknown' symbol for the training, and for generating output, each 'unknown' symbol was replaced by the word with the highest probability score from the attention layer. However, no scripts or guidelines were provided for it. Also, it was not mentioned that the words were segmented into sub-word units, which is nowadays the standard for all state-of-the-art neural systems. Word segmentation enables better coverage of large vocabularies and treatment

		original	**simplified**
train	*sentences*	284,677	
	words	7,401,589	5,635,507
	vocabulary	212,292	165,170
test	*sentences*	360	
	words	8,110	7,957
	vocabulary	3,209	2,802

Table 1: Data set statistics showing the number of sentence pairs in the training and test set, and the number of words and vocabulary size for the non-simplified and simplified versions of sentences separately.

of rare and unseen words. The standard word segmentation method for the Sockeye tool is byte-pair encoding (BPE) (Sennrich et al., 2016), which is one of the most widely used segmentation methods. According to the Sockeye guidelines, segmentation is performed after the original text is tokenised. In our experiments, we explored both original and additionally tokenised data, both with BPE word segmentation.

After generating outputs with our transformer models, sub-word units are joined together to form original words. This is usually followed by a detokenisation step. However, since the outputs of the original models are all tokenised, we evaluated both versions: tokenised and detokenised. Finally, due to lack of special treatment of named entities, the transformer outputs contain a number of 'unknown' symbols, referring to unseen sub-word units. We computed metric scores for two versions of the output: with 'unknown' symbols left in place, and with 'unknown' symbols removed.

2.4 Evaluation

We performed automatic evaluation of generated outputs using the script provided by the authors which calculates two metrics: BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016). Previous work also explored differences arising from different BLEU implementations (Popović and Belz, 2021), but these are not relevant to present purposes. BLEU is based on matching between the generated text and a manually simplified reference text, while SARI compares the generated text both to the reference text as well as to the original text.

2.5 Methodological, Arbitrary and Incidental Variations

Table 2 provides an overview of the experimental conditions (first column) for which we explored

Condition	Explanation	Range of values explored
Methodological variation		
Model type	different types of neural architectures	LSTM, transformer
Implementation	different implementations of same model	two versions of the Sockeye tool
Preprocessing	different types of text processing	word segmentation, tokenisation, treatment of named entities
Arbitrary variation		
Random seed	weights initialization, data order, and sampling used in generation	36 different initialisations
Incidental variation		
Dependency versions	changes in external libraries/dependencies	Python, Lua, NLTK
Run-time environment	where/when the experiment was carried out	evaluation, test, training

Table 2: Summary of different experimental conditions explored in our runs (see in text for explanation of three broad categories).

variation. The conditions are grouped into three categories: (i) methodological factors, i.e., variation in the methods used in a solution for a task with the aim of improving performance, where better performance can to some degree be expected to generalise to similar types of tasks; (ii) arbitrary factors where an arbitrary (often random) selection is made with respect to a given parameter; and (iii) incidental factors, where selection is not under the direct control of the system creators, e.g., changes from one version of a dependency to another. All of these conditions may be reasonably expected to vary during replication experiments.

Methodological factors may occur when the group replicating a given model decides to update some component of its design based on recent findings. An example in our own work reported here is the inclusion of the transformer-based model, based on the recent success of these models for a wide range of NLP tasks in the time since Nisioi et al. (2017)'s publication.

Arbitrary factors may occur due to underreporting of necessary parameters in the original work. For instance, if a hyper-parameter must be specified in order for the model to run but no specifications are provided by the model creators, the group replicating the work may select that hyper-parameter randomly or using their own heuristic.

Incidental factors may occur due to library or package updates, rendering the versions reported in the original publication obsolete. It also may occur in different run-time environments, for example running experiments on different computers.

By including each of these factors in our study, we sought to ensure broad coverage of the range of results variation that may realistically occur when attempting to replicate a previously reported model.

3 Results

We report the results from both team A and team B, for each of the studied conditions. While both teams struggled to get the original repository to a working state, team A failed to install all the required dependencies as many are deprecated. Team B reported similar concerns about reproducing and reusing the original source code; however, ultimately, they managed to get the repository to a running state.

Table 3 shows the two automatic scores generated by the evaluation script provided by the authors for all explored variations (see Table 2), grouped together by system: LSTM, LSTM-w2v, Transformer Sockeye v1 and Transformer Sockeye v3. Where they exist, results provided by the authors of the original paper are included as well. For random seed search, we included two worst and best-performing models in this table, while full results of this search can be found in Appendix.

Averaged scores for each of the three models together with the standard deviations and coefficients of variation (Belz et al., 2022a) are presented in Table 4. For each of the models, 'all' refers to the average value of all scores for this model presented in Table 3. For the LSTM model, 'random seed' is averaged only over the random seed scores, and 'other' is averaged over all scores except the random seed scores. For the transformer model, 'v1' means only the scores from version 1, and 'v3' means only the scores from version 3.

System	Training		Outputs		Scores (original script)		
	trained by	*on data set*	*generated by*	*post-processing*	*SARI*	*BLEU*	*run by*
LSTM/ OpenNMT	N et al, 2017	original	N et al, 2017	original	30.65	84.51	N et al, 2017
	N et al, 2017	original	N et al, 2017	original	30.65	85.60	team A, 2022
	N et al, 2017	original	N et al, 2017	original	30.65	84.51	team B, 2022
	N et al, 2017	original	team A, 2021	original	29.96	86.61	team A, 2022
	N et al, 2017	original	team B, 2022	original	29.96	86.53	team B, 2022
	team B, 2022	original	team B, 2022	original	30.23	88.81	team B, 2022
	team B, 2022	original	team B, 2022	original	28.68	84.47 ‡	team B, 2022
	team B, 2022	original	team B, 2022	original	29.76	89.59 †	team B, 2022
	team B, 2023	original	team B, 2023	original	29.53	88.68	team B, 2023
LSTM-w2v/ OpenNMT	N et al, 2017	original	N et al, 2017	original	31.11	87.50	N et al, 2017
	N et al, 2017	original	N et al, 2017	original	31.11	89.36	team A, 2022
	N et al, 2017	original	N et al, 2017	original	31.11	87.50	team B, 2022
	N et al, 2017	original	team A, 2021	original	29.12	89.64	team A, 2022
	N et al, 2017	original	team B, 2022	original	29.12	89.40	team B, 2022
	team B, 2022	original	team B, 2022	original	29.70	87.04	team B, 2022
	team B, 2023	original	team B, 2023	original	29.74	88.56	team B, 2023
Transformer/ Sockeye v1 (MXNet)	team A, 2022	original+BPE	team A, 2022	BPE joined	32.67	84.66	team A, 2022
				+'unk' removed	32.67	89.75	
				+detokenised	32.64	84.00	
				+detok+'unk'	32.70	88.45	
	team A, 2022	tokenise+BPE	team A, 2022	BPE joined	32.54	80.32	team A, 2022
				+'unk' removed	32.54	86.15	
				+detokenised	32.86	83.52	
				+detok+'unk'	32.90	88.55	
Transformer/ Sockeye v3 (PyTorch)	team A, 2022	original+BPE	team A, 2022	BPE joined	28.41	91.82	team A, 2022
				+'unk' removed	28.40	93.74	
				+detokenised	32.66	90.95	
				+detok+'unk'	32.70	92.45	
	team A, 2022	tokenise+BPE	team A, 2022	BPE joined	29.50	88.30	team A, 2022
				+'unk' removed	29.49	89.97	
				+detokenised	32.94	91.00	
				+detok+'unk'	32.94	91.72	

Table 3: BLEU and SARI scores for different experimental variations. † is the best-performing model in the random seed search, ‡ is the worst performing model in the random seed search.

According to the averaged SARI score, the transformer model performs best; however, the newest version performs worse than the old one. According to the averaged BLEU score,[6] LSTM-v2w and Transformer have very similar performance, but the newest version of the transformer is the best of all while the first version is the worst.

We used the R package *cvequality* (Version 0.2.0; (Marwick and Krishnamoorthy, 2019)) to test for significant differences of coefficients of variation (CV). This package implements two of the most widely used statistical significance tests, proposed by Feltz and Miller (1996) and Krishnamoorthy and Lee (2014). The null hypothesis for each of the two automatic metrics is that there is no difference in CV between the three models.

We use the results reported in the Table 4 corresponding to the row 'all' for the three model variants. Conducting the two tests resulted in the statistical significance values shown in Table 5. We observe that neither test statistics nor p-value suggest statistical significance when setting $\alpha = 0.05$. Therefore, we cannot reject the null hypothesis.

4 Discussion

Nisioi et al. (2017) reported that using pre-trained word embeddings improves the model's performance. Results in Table 3 and Table 4 suggest that while this may be true, the differences are too small to draw clear conclusions. For one model alone, the LSTM variant, we have observed BLEU scores ranging from 84.47 to 89.59; the average, on the other hand, is 87.90 with the CV of 1.36. Compared to LSTMs, transformer models have a higher variance in their performance. This can be attributed to the transformer's complexity and the fact that they are harder to train. Also, variations in tokenisation were included only in the transformer models. The performance difference between the best and worst transformer models is even higher

[6]The reason for slightly different scores on original outputs is yet another source of variation which we did not explore here, namely incidental variations of BLEU scores related to dependencies and run-time environment.

		SARI			BLEU		
model		*avg.*	*dev.*	*CV*	*avg.*	*dev.*	*CV*
LSTM	all	29.38	0.48	1.66	87.64	1.39	1.59
	random seed	29.24	0.31	1.07	87.90	1.18	1.36
	other	30.23	0.51	1.74	86.07	1.66	2.00
LSTM-w2v	all	30.14	0.98	3.35	88.43	1.12	1.31
transformer	all	31.78	1.75	5.58	88.47	3.83	4.40
	v1	32.69	0.13	0.43	85.71	3.32	3.99
	v3	30.88	2.18	7.29	91.24	1.69	1.91

Table 4: Average SARI and BLEU scores, standard deviations and coefficients of variation (CV) for the three models.

test	**BLEU**	**SARI**
Feltz & Miller	3.54 / 0.16	2.98 / 0.22
Krishnamoorthy & Lee	1.72 / 0.42	1.59 / 0.44

Table 5: Test statistics / p-value are reported for differences between coefficients of variation (CVs) of the three models reported in Table 4, for both BLEU and SARI.

than LSTM variants. With a 13.42 BLEU score difference, assessing *true* performance of the model is a challenging task. Judging the results by the average BLEU score (Table 4), we can observe that the transformer model trained using v3 of the Sockeye tool outperforms the rest of the models. This model achieves an average BLEU of 91.24 with a CV of 1.91. To put the CV into context, this value is higher than three other LSTM variants but lower than the rest of the transformer models. As it can be expected, using an averaged performance metric and CV enables a better comparison between models in different conditions.

Besides the mentioned analysis, we found it hard to provide distinct and unique observations from the results. This is likely due to the fact that the results are not conclusive and the variance is high. We do not believe this is a flaw in our experimental design but rather a good representation of the complexities of comparing different models across varying conditions. The number of experiments conducted in this study is more than 60, a number that exceeds the number of experiments conducted in most other studies by a large margin.

One of the concerning issues we encountered is the issue of software deprecation. While this is not a new problem, and it is as old as software itself, it is becoming more and more prevalent. This is due to extreme reliance on empirical results and the complexity of publications that utilise neural networks. Often source codes use several external libraries and dependencies, any of which may become deprecated at any time. Increased availability of source code and the abundance of tools are signs of a healthy research community. Seeing new tools and libraries developed and improved daily is encouraging. At the same time, we believe researchers should practice caution when introducing new tools and libraries into their experiments, as doing so may shorten the usability of their source code.

4.1 Addressing Experimental Variation in Experimental Design

Many factors can affect the results of an experiment. Some of these factors are under the experimenter's control, and some are not. Before we address these variations, we highlight that scientific experiments are developed as a counterpart to abstraction of real-world problems. Data sets are created with this in mind, consisting of training, validation, and test sets of which the latter, in particular, is created to represent unseen real-world data. Research on improving the generalisation of machine learning algorithms is another good example of leveraging scientific experiments to understand real-world challenges.

We can use another analogy to explore these variations further. `Bogosort` is a sorting algorithm that generates random permutations of the input until the input is sorted. While in the best case, it may take $O(n)$ steps to sort the input, its worst-case performance is unbounded, making it impractical to use. Theoretically, it is possible to find the random seed that achieves best-case performance for a specific input; nonetheless, the slightest change in

hardware, environment, or even the input itself will render this seed useless. Although neural networks are far more complicated than a simple sorting algorithm, the basis of reliance on the evidence is the same. Similar to `Bogosort`, recording all the random numbers used in an experiment is possible (Chen et al., 2022a), but the question is: should we? We do not think so. Instead of optimising the random seed or other arbitrary factors, researchers should focus on the methods that minimize the impact of these variables. Ultimately, we believe the correct approach for conducting scientific experiments is to thoroughly report methodological variations, control incidental variations, and abstract away arbitrary variations.

5 Conclusions

In this work, we conducted a series of experiments for a single task using the same data under different experimental conditions. We categorized these conditions into three different categories: methodological, arbitrary, and incidental. We report the results of our experiments to demonstrate the wide results variation that can occur due to these factors.

We propose that researchers should record all methodological conditions, control incidental ones, and abstract away arbitrary factors. Lastly, we observed that using average score and its coefficient of variation (CV) instead of the maximum value provides far more reliable results. We recommend that researchers adopt this practice when documenting the findings from their own studies.

We are aware that this is easier said than done. We are, however, optimistic that the field can move closer to this ideal over time. In the meantime, it is our hope that this recommendation highlights the contrast between what is currently a common practice (unfortunately, inadequate recording and reporting that do not address necessary factors for reproducibility) and what is needed to support successful, reproducible research in our field.

Limitations

Our work is limited by several factors. First, our findings are supported only by experiments on a single NLP task (neural text simplification). We selected this task because it offered an intriguing sandbox for studying varying experimental conditions, ranging from differences in random seeds to modifications in compile-time and run-time environments and dependency versions. Comparing the multifaceted outcomes arising from these experiments facilitated greater quantified estimations of the degree of reproducibility for the selected NTS systems. However, the dimensions of variation that we explored in this work are common to many NLP tasks; none are unique only to text simplification. Because of this, we believe that our findings would generalise broadly across NLP tasks.

We used a single data set, the same as in the original paper by Nisioi et al. (2017), to foster controlled study of our other experimental variables. The data set comprises aligned sentences between English Wikipedia and Simple English Wikipedia. Thus, it is unclear whether our findings would be similar if the study was conducted using data from other languages, including those with richer morphology such as Czech or Arabic.

Finally, although we conducted a robust set of experiments for the selected models across two research groups, our experiments are limited to a small set of NTS models due to the extensive set of conditions tested for each model. Although these models vary in their architecture, we do not know if other NTS models may be more or less stable across experimental conditions. Taken together, the limitations accompanying our findings suggest compelling avenues for future research.

Ethics Statement

This research was guided by a broad range of ethical considerations, taking into account factors associated with environmental impact, equitable access, and reproducibility. We summarize those that we consider most critical in this section. It is our hope that by building a holistic understanding of these factors, we develop improved perspective of the challenges associated with reproducibility studies and the positive broader impacts that improved reproducibility standards may promote.

Environmental Impact. In this work, we seek to study the complex and murky relationship between experimental conditions and experimental outcomes. To address research questions surrounding this relationship, we conduct many experimental runs to replicate the same models across an extensive set of variable conditions. Although necessary for justifying our claims, a downside of this process is that it may produce environmental harm. One might argue that the advantages of assurance that the 'true' evaluation score is found do not outweigh the disadvantages of repeatedly

running models that are known to produce large carbon footprints (Strubell et al., 2019). We attenuate this risk by controlling for as many variables allowable (e.g., data set and architectural variations) while still fostering robust study of our core question, to minimize the number of experimental runs required.

Equitable Access. A concern closely related to environmental impact is that of equitable access to this line of research. By studying a problem that requires many repeated experimental runs with subtle variations, we may exclude disadvantaged researchers from performing meaningful follow-up studies, since they may not have the requisite resource bandwidth (Bommasani et al., 2021, §5.6). However, although reproducibility studies themselves may pose a barrier to entry for researchers with limited access to compute hardware, the innovations *resulting* from these studies (e.g., improved community standards for reproducibility of reported results) may stand to greatly benefit marginalised researchers, by minimising the potential for bottlenecks in attempting to perform impossible and costly replications to establish performance baselines.

Reproducibility. To ensure reproducibility of our own work, we report all experimental parameters, computational budget, and computing infrastructure used. We discuss our experimental setups in depth, as they are the primary focus of this study. We report descriptive statistics about our results to enhance transparency of our findings, and we report all implementation settings (e.g., package version number) needed to successfully replicate our work. Although reproducibility studies are not specified as an intended use of the referenced systems (Nisioi et al., 2017), this use is compatible with the original access conditions and the authors have consented to the paper's use in numerous reproducibility studies since its publication (Belz et al., 2022b).

Acknowledgements

This research was conducted with the financial support of Science Foundation Ireland under Grant Agreement No. 13/RC/2106_P2 at the ADAPT SFI Research Centre at Dublin City University. ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology, is funded by Science Foundation Ireland through the SFI Research Centres Programme.

References

Mohammad Arvan, Luís Pina, and Natalie Parde. 2022a. Reproducibility in computational linguistics: Is source code enough? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2350–2361, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mohammad Arvan, Luís Pina, and Natalie Parde. 2022b. Reproducibility of *Exploring Neural Text Simplification Models*: A Review. In *Proceedings of the 15th International Natural Language Generation Conference (INLG 2022)*, Waterville, ME.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2020. ReproGen: Proposal for a shared task on reproducibility of human evaluations in NLG. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 232–236, Dublin, Ireland. Association for Computational Linguistics.

Anya Belz, Maja Popovic, and Simon Mille. 2022a. 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*, pages 249–258, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Anya Belz, Anastasia Shimorina, Maja Popović, and Ehud Reiter. 2022b. The 2022 reprogen shared task on reproducibility of evaluations in nlg: Overview and results. In *Proceedings of the 2022 ReproGen Shared Task on Reproducibility of Evaluations in NLG (ReproGen 2022)*, pages 1–9, Waterville, Maine. Association for Computational Linguistics.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent,

Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. *ArXiv*.

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 Twelfth Language Resources and Evaluation Conference*, pages 5539–5545, Marseille, France. European Language Resources Association.

Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors. 2022. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States.

Boyuan Chen, Mingzhi Wen, Yong Shi, Dayi Lin, Gopi Krishnan Rajbahadur, and Zhen Ming Jiang. 2022a. Towards training reproducible deep learning models. In *44th IEEE/ACM 44th International Conference on Software Engineering, ICSE 2022, Pittsburgh, PA, USA, May 25-27, 2022*, pages 2202–2214. ACM.

Yanran Chen, Jonas Belouadi, and Steffen Eger. 2022b. Reproducibility issues for bert-based evaluation metrics. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2965–2989, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Michael Cooper and Matthew Shardlow. 2020. CombiNMT: An exploration into neural text simplification models. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5588–5594, Marseille, France. European Language Resources Association.

Carol J Feltz and G Edward Miller. 1996. An asymptotic test for the equality of coefficients of variation from k populations. *Statistics in medicine*, 15(6):647–658.

Felix Hieber, Michael Denkowski, Tobias Domhan, Barbara Darques Barros, Celina Dong Ye, Xing Niu, Cuong Hoang, Ke Tran, Benjamin Hsu, Maria Nadejde, Surafel Lakew, Prashant Mathur, Anna Currey, and Marcello Federico. 2022. Sockeye 3: Fast Neural Machine Translation with PyTorch. *arXiv preprint https://arxiv.org/abs/2207.05851v4*.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2018. The sockeye neural machine translation toolkit at AMTA 2018. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 200–207, Boston, MA. Association for Machine Translation in the Americas.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning sentences from standard Wikipedia to Simple Wikipedia. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 211–217, Denver, Colorado. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Kalimuthu Krishnamoorthy and Meesook Lee. 2014. Improved tests for the equality of normal coefficients of variation. *Computational statistics*, 29:215–232.

Ben Marwick and Kalimuthu Krishnamoorthy. 2019. cvequality: Tests for the equality of coefficients of variation from multiple groups. R package version 0.2.0.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

Maja Popović and Anya Belz. 2021. A reproduction study of an annotation-based human evaluation of MT outputs. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 293–300, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Maja Popović, Rudali Huidrom, Sheila Castilho, and Anya Belz. 2022. Reproducing a manual evaluation of simplicity in text simplification system outputs. In *International Natural Language Generation Conference (INLG 2022)*. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Soren Sonnenburg, Mikio L Braun, Cheng Soon Ong, Samy Bengio, Leon Bottou, Geoffrey Holmes, Yann LeCunn, Klaus-Robert Muller, Fernando Pereira, Carl Edward Rasmussen, et al. 2007. The need for open source software in machine learning. *Journal of Machine Learning Research*, 8:2443–2466.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS 2017)*, pages 5998–6008, Long Beach, CA.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

A LSTM Random Seed Search

We provide the full experimental results from the random seed search in this appendix. For each variant, we include its perplexity, SARI, and BLEU score.

Variant	Perplexity	SARI	BLEU
nts_search_3	10.30	28.68	84.47
nts_search_14	10.49	28.62	85.04
nts_search_24	10.25	28.94	85.28
nts_search_10	10.26	28.88	86.69
nts_search_16	10.45	29.60	86.81
nts_search_20	10.22	29.02	86.95
nts_search_4	10.63	29.78	87.14
nts_search_2	10.27	29.34	87.19
nts_search_31	10.34	29.31	87.21
nts_search_17	10.13	29.40	87.42
nts_search_23	10.31	29.19	87.51
nts_search_0	10.37	28.95	87.75
nts_search_15	10.33	28.96	87.77
nts_search_25	10.21	29.62	87.81
nts_search_39	10.24	28.83	87.81
nts_search_38	10.32	29.28	87.84
nts_search_36	10.29	29.11	87.86
nts_search_33	10.39	28.99	87.94
nts_search_22	10.32	29.02	88.28
nts_search_37	10.20	29.24	88.36
nts_search_29	10.33	29.31	88.42
nts_search_26	10.16	29.18	88.42
nts_search_1	10.32	29.17	88.58
nts_search_32	10.24	29.15	88.59
nts_search_19	10.43	29.29	88.61
nts_search_18	10.49	29.50	88.64
nts_search_11	10.30	28.98	88.68
nts_search_12	10.24	29.55	88.69
nts_search_21	10.30	29.89	88.75
nts_search_41	10.38	29.32	88.83
nts_search_13	10.12	29.59	88.98
nts_search_35	10.39	29.14	89.01
nts_search_34	10.34	29.30	89.03
nts_search_28	10.34	29.15	89.14
nts_search_30	10.16	29.71	89.42
nts_search_27	10.21	29.76	89.59

Table 6: Results of the full random seed search, with perplexity, SARI, and BLEU scores reported for each variant.

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

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

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ 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)?
Left blank.

☐ 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?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ 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.
Left blank.

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?
Left blank.

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

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ 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?
Left blank.

☐ 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.)?
Left blank.

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

Left blank.

☐ 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.?
Left blank.

☐ 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)?
Left blank.

☐ 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?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.