(Mostly) Automatic Experiment Execution for Human Evaluations of NLP Systems

Craig Thomson
ADAPT/DCU, Ireland
craig.thomson@dcu.ie

Anya Belz
ADAPT/DCU, Ireland
anya.belz@adaptcentre.ie

Abstract

Human evaluation is widely considered the most reliable form of evaluation in NLP, but recent research has shown it to be riddled with mistakes, often as a result of manual execution of tasks. This paper argues that such mistakes could be avoided if we were to automate, as much as is practical, the process of performing experiments for human evaluation of NLP systems. We provide a simple methodology that can improve both the transparency and reproducibility of experiments. We show how the sequence of component processes of a human evaluation can be defined in advance, facilitating full or partial automation, detailed pre-registration of the process, and research transparency and repeatability.

1 Introduction

The traditional method for recording the steps performed in a scientific experiment is the pen and paper logbook. Barker (1998) argues that in the event of a fire in the lab, it is the only thing that one should grab, leaving computers, physical samples, and expensive equipment behind. In fields such as chemistry, students are taught systematic approaches for completing such records, which commonly include the date of the experiment, the hypothesis, the steps carried out, and the results.[1,2]

These days, researchers may feel less compelled to grab their paper records (or even their computer) in case of fire, since they can record their notebooks digitally and have them immediately backed up to the cloud. However, at least in Natural Language Processing (NLP), it appears that this has not helped to ensure survival of records of experimental procedures which are rarely available after the fact, in any form (Belz et al., 2023a,b). Even basic records and other data files such as the set of system outputs that were evaluated or the question that participants were asked are seldom made publicly available (Belz et al., 2023b). When contacted, around two thirds of corresponding authors do not respond (Belz et al., 2023a), and only around half of those who do can provide this basic information. Mistakes by researchers whilst running experiments are depressingly common (Thomson et al., 2024) and reproduction attempts often struggle to find and follow the original procedure, even with the help of the authors (Arvan and Parde, 2023; Li et al., 2023; van Miltenburg et al., 2023).

Automated experimentation techniques (Robertson et al., 2009), where the experimental process is defined in advance and researcher intervention kept to a minimum during experiment execution, can remove reliance upon error-prone manual data entry. Such techniques also benefit from having a clear experimental procedure which must be defined in advance, making it impossible for researchers to change the configuration part way through a run (accidentally or nefariously). Automating processes is essential for large scale experiments where massive volumes of data are collected and processed in real time, e.g., in particle physics (Gaspar et al., 2021). For the field of Economics, Gentzkow and Shapiro (2014) propose that researchers should automate everything they can, ideally with a single code script, such that repeatability is ensured.

The state of human evaluation in NLP research more generally is dire (Gehrmann et al., 2023). Most work reporting on the state of human evaluation in NLP research has focused on aspects of design such as participant guidelines (Ruan et al., 2024), quality criterion names and definitions assessed (Howcroft et al., 2020), or the comparability of experiments (Belz et al., 2020). Such aspects of the experimental design are vitally important, but separate to the question of how the experiment

[1] https://libguides.wpi.edu/ch1010/lab_notebooks

[2] https://web.stanford.edu/class/chem184/manual/LabNotebook.pdf

Proceedings of the 17th International Natural Language Generation Conference, pages 272–279
September 23–27, 2024. ©2024 Association for Computational Linguistics

procedure is recorded and executed.

We argue that many of the above issues would be at least ameliorated by automating experimental execution as much as possible. Some experiments, such as those that use crowd platforms like Amazon Mechanical Turk or Prolific, can be fully automated using the available APIs. At a minimum, it is straightforward to see from Figure 1 that everything prior to *Present Participants with Evaluation Items* can be automated as one pipeline, as can everything from *Responses* onwards. In both cases, we would simply be pipelining a series of operations on data. Automation can also be applied to the process of collecting responses and checking/excluding them.[3]

In the rest of this paper, we start by investigating whether the individual files and component processes that make up a human evaluation tend to be reported (Section 2), before proposing a methodology for achieving automation (Section 3). We describe an example application of the methodology (Section 4) and end with some conclusions and a look to future work (Section 5).

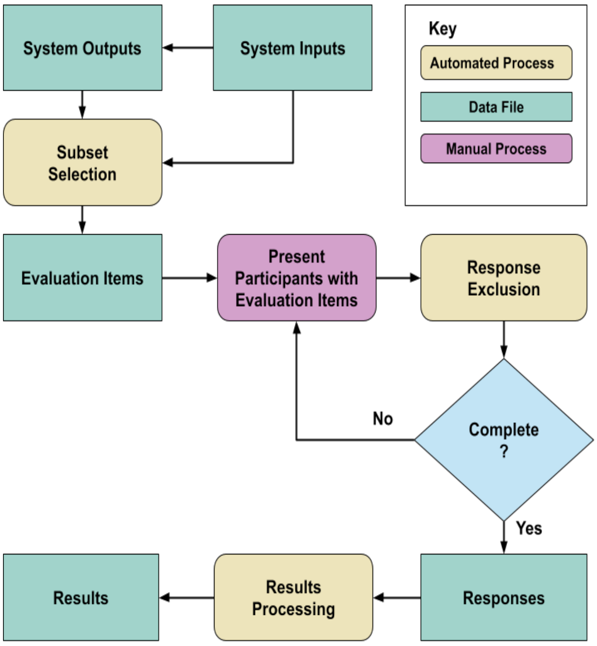

Figure 1: Diagram showing the flow control of the notebook used to demonstrate the proposed approach to automating human evaluation experiments. All steps except for presenting the evaluation items to participants are simple to automate before the experiment.

2 Availability of Experiment Components

We performed a systematic analysis of papers made available in the ReproNLP 2024 shared task on reproducibility of evaluations in NLP (Belz and Thomson, 2024), with the aim of establishing which evaluation experiment components were (not) made available by researchers. The shared tasks organisers made available resources that were obtained from the authors, including the evaluation items and interface. With only 5% of authors making such details publicly available (Belz et al., 2023b) and only 17% of authors being able to do so after being contacted (Belz et al., 2023a), ReproNLP provides a good sample of 20 papers where authors have made the effort to share resources.

We broke down the experimental process into the data files and component processes shown in Figure 1. Rather than use a more complex process with exhaustive options that cover all types of human evaluation, we use the simplest overall process that includes exclusion of responses. We argue that most human evaluations of NLP system quality will require these component processes, even if they also include other ones or the control flow logic differs (for a more generally applicable breakdown into component processes see Belz et al. (2024)). It therefore is a good vanilla design that is useful for both designing experiments, and for checking that published papers include at least minimal data files and component process definitions. Note that *Response Exclusion* needs to be handled with care and should always be fully specified in advance.

We then annotated each paper, first checking to ensure that the overall process shown in Figure 1 was applicable to the experiment being carried out in the paper (it was in all cases). We then checked files and component process definitions were available. When doing so, we looked only for evidence of the resources being available; we did not check their validity.

Anonymised results of our annotation process are shown in Table 1. We found that only 4 of 20 papers made available the complete set of *System Inputs*, *System Outputs* and the *Subset Selection* process by which *Evaluation Items* were created from them. Whilst 12 of the 20 papers provided the participant *Responses*, only four of those provided scripts for *Results Processing*, with only two of those performing statistical tests. Of the six papers where *Response Exclusion* was performed, the process was not recorded in any of them. We also

[3] An example of how to do this is included in our example experiment on GitHub: `https://github.com/nlgcat/mostly_automated`.

Table 1: Matrix showing what information (data or component process definition) was available for each anonymised paper (lettered A–T). The cell contents key is as follows: y => yes (was available), n => no (was not available) x => not applicable (the paper explicitly indicated this process/data was not part of the experiment), and u => unknown (we could not tell whether the process/data was meant to be part of the experiment).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
System Inputs	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
System Outputs	y	y	n	n	y	n	n	y	y	y	n	n	n	n	n	y	y	n	y	y
Subset Selection	y	y	y	n	n	n	n	n	n	n	n	n	n	n	n	y	n	n	y	n
Evaluation Items	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
Response Exclusion	n	u	u	n	u	u	u	x	u	u	x	n	u	u	u	n	u	u	u	n
Responses	y	n	y	n	n	y	n	y	y	y	n	n	y	y	y	n	y	n	y	y
Results Processing	y	n	y	n	n	n	n	y	n	n	n	n	n	n	n	y	n	n	y	n
Statistical Analysis	y	x	x	x	n	n	n	x	n	x	n	n	n	n	n	y	n	x	y	n
Results	n	n	y	n	n	n	n	y	n	n	n	n	n	n	n	n	n	n	y	n

noted that whilst *Results* are presented in all papers, only 3 of 20 included structured data files containing the same results as tables in the paper. Whilst all papers shared *Evaluation Items*, this was a prerequisite of selection for the ReproHum project. All papers sharing *System Inputs* tended to mean the dataset used was cited in the paper, even if the system inputs were not included directly in the supplementary material.

Many of the human evaluations of NLP systems in the literature are not very complex in terms of the overall process. Most experiments are comparisons of a small set of systems, and ask participants to directly assess or compare texts on simple questionnaire forms. Such simple experiments can be easily automated, especially by computer scientists.

3 Proposed Methodology

We propose a simple and flexible high-level methodology for creating mostly-automated experiments that evaluate the performance of NLP systems. For the sake of brevity, below we refer to these simply as human evaluations, with the caveat that the vanilla experiment structure introduced above is likely not applicable to all experiments.

The overall procedure for human evaluation experiments can be broken down into component processes, where each component process takes one or more data files as input, performs some operation on them, and then outputs one or more data files. Figure 1 shows an example of a minimal human evaluation experiment modelled in this way. An example component process is to input the *System Inputs* and *System Outputs* to the *Subset Selection* component process that outputs the *Evaluation Items*. The flow control of the overall process can then be modelled with simple loops and other conditional logic, using basic computing science concepts. Processes may be fully automated, partially automated, or entirely manual. The crucial thing is that they are fully defined in advance of the experiment and then automated as much as possible.

Most experiments will require additional steps in practice, although, with the exception of *Response Exclusion*, those shown in Figure 1 are core steps that most human evaluations would require in order to function at all. For examples of similar but more fine-grained methods designed for the similar task of dataset annotation, please see Oortwijn et al. (2021) and Klie et al. (2024). Each of the files and processes in Figure 1 can be mapped to a question in the Human Evaluation Datasheet (HEDS), which includes comprehensive details of possible components in a human evaluation (Shimorina and Belz, 2022).

Note that we do not consider here the design of the interface or questions that participants are asked. These are important considerations but separate from issues of process.

3.1 Subset selection / distribution of evaluation items

There are different methods by which subsets of evaluation items can be selected, for example, randomly, or by stratified sampling. In terms of reproducibility, the important thing is that the process is recorded in a deterministic way.

3.2 Exclusion of responses

It is bad practice to define the process by which responses are excluded, for whatever reason, *after* the participant responses have been seen as it introduces researcher bias (Thomson et al., 2024). It is also important to record the process for excluding responses, otherwise it can be difficult to reproduce (Arvan and Parde, 2023; González Corbelle et al., 2023; van Miltenburg et al., 2023; Watson and Gkatzia, 2023). Since the process can and should be defined in advance, it can be implemented as a script

3.3 Presenting evaluation items to participants

Whilst it is possible to automate this component process, i.e., by automatically posting a survey online or using APIs from crowd-sourcing platforms such as Amazon Mechanical Turk or Prolific, there might be some cases where it is impractical to do so and participants will need to be given forms manually by the researcher. For example, if each participant needs to complete a spreadsheet, or if the researcher is configuring an experiment on the web interface of a crowd-source platform.

However, component processes as described in Section 3 can still be used. Input files (such as forms, data, and spreadsheets) must still be processed (given to participants so they can record their responses). The crucial thing is that the process by which the researcher interacts with participants is minimised and clearly documented in advance. Any person with strong administrative skills could then execute this part of the experiment (they need not know the details of the design, only the steps required to run it).

3.4 Collating annotated evaluation items

Collating the files from the previous component process can be fully automated. The files should be in a known format, with clear names that include prefixes for things such as the participant ID. Tests can then be written to confirm that all evaluation items have the correct number of judgments. If any work is to be repeated, e.g. due to failed attention checks, the system should create the required files and instruct the researcher such that they can present them to the participants, reducing the manual work the researcher is performing during the experiment, with the aim of reducing mistakes. This loop is repeated until a complete set of valid responses is obtained.[4]

3.5 Results processing

The required type(s) of statistical analysis should be determined as part of the experiment design process, in advance of the experiment. This could be implemented e.g. with simple conditional logic such as selecting parametric or non-parametric tests based on the distribution of the results. Since the format of the data files containing evaluation items and participant responses are also known, the statistical analysis code can be written in advance.

3.6 Post hoc analyses

Post hoc analyses are a valid method of data analysis after the conclusion of an experiment. Indeed, they are often vital in improving our understanding of the data and in designing future experiments. However, they should be clearly identified as post hoc and performed as additional steps at the end end of an experiment, without changing the existing procedure or code.

3.7 Dummy experiments

Once the evaluation items have been selected and distributed into per-participant lists, and the hypothesis has been defined, it is possible to perform a dummy run of the entire experiment. Automatically generated results, following both normal and random distributions, can be used in place of participant responses, allowing for the downstream process to be tested in advance.

4 Example experiment

In this section we describe an example experiment where data-to-text system outputs are evaluated. For this, we use data and system outputs from the WebNLG 2017 Challenge (Gardent et al., 2017), where systems convert structured input (triples) to text. The entire experiment is encoded in a Jupyter notebook which is included as supplementary material. For system texts we use the constant string "Example Text" since we are not showing any texts to participants during our implementation.

4.1 Subset selection/distribution

Items in the WebNLG dataset can be grouped by category (Airport, Building., etc.) and number of triples (1-7). For this experiment we will be evaluating the performance of systems from the Airport, Building, and City categories, for triples sizes of between 1 and 4. Note that this is an arbitrary design choice and is not representative of the entire dataset. As with all of our examples, it is illustrative, and the important thing is that we encode what we are doing. We will use stratified sampling to select 15 input items, with three system outputs (including one human authored reference) for each of the 12 property combinations (category×size),

[4] See the Jupyter notebook at https://github.com/nlgcat/mostly_automated for an example of how this can be implemented.

six participants will then be asked to rate each item, with each participant rating 36 items (1 of each property combination for each system). This experiment will therefore require a total of 3,240 total judgments, obtained from 90 distinct participants. The experiment is designed to be run on Amazon Mechanical Turk.

4.2 Response exclusion

We exclude responses from any participant who responded with the same score for each of the 36 outputs they rate. Note that this is a weak exclusion criterion, used only for illustrative purposes.

4.3 Presenting items

Amazon Mechanical Turk requires a CSV file that is used to populate an HTML form template. Each row of the CSV file represents a list of evaluation items containing all 36 evaluation items that will be shown a participant, with multiple sets of columns representing the system input, output, and meta data for each evaluation item.[5] Our code must take the output of Section 4.1 and prepare the input CSV file. Finally, with minimal manual intervention, the researcher will then configure MTurk. Note that this could be entirely automated using deployment scripts and the MTurk API, although we illustrate here that some manual intervention can still be part of the experiment, provided that a procedure for the researcher to follow is clearly defined in advance. Not all researchers will have the time or ability to perform complex software engineering deployments.

4.4 Collating results

Amazon Mechanical Turk outputs a CSV file in the same format as its input file, with the addition of response values and meta data. If any of the responses are missing or invalid due to predefined attention checks, our code processes only the valid response rows, and creates a file containing rows that need to be repeated by one or more additional participants so that the researcher can upload that to Mechanical Turk to obtain replacement responses.

4.5 Results processing

Our null hypothesis is that there is no difference between the selected systems in terms of level of grammaticality. If results are normally distributed, as determined by the Shapiro-Wilk test (Shapiro and Wilk, 1965), then we will use an Anova, if not, a Kruskal-Wallis test (Kruskal and Wallis, 1952). If there is a significant result we will also perform pairwise T-tests or Wilcoxon signed-rank tests as appropriate with α being set to 0.05. Inter-annotator agreement will also be calculated using Krippendorff's Alpha (Krippendorff, 2004) in ordinal mode. A threshold of 0.67 is set for tentative conclusions, and 0.8 to deem our results reliable. We also create code to trivially add results tables and figures to our paper.

4.6 Dummy experiments

Three types of dummy responses were created for testing; *Random*, where each response was random, *Static*, where each system is always given the same score {A=>2, B=3, C=4}, and *Normal*, where normal distributions are created around a mean taken as the *Static* score with a standard deviation of 1.0. Figures 2–4 in the appendix show stacked bar charts of these distributions. Table 2 shows some example results from the dummy responses. As expected, Static and Normal have significant differences between populations, but only static has strong inter-annotator agreement (participant responses within Dist are randomly taken from the normal distribution).

Table 2: Results of the Kruskal-Wallis and Krippendorff's α (ordinal method) tests for the different types of dummy response distribution. Note that p-values for *Static* and *Normal* are infinitesimal.

	Kruskal-Wallis		
Distribution Type	F-statistic	p-value	K's α
Random	0.89	0.64	0.01
Static	3239.00	< 0.001	1.00
Normal	1282.57	< 0.001	0.40

5 Conclusion and Future Work

Many of the suggestions we make in this paper may seem obvious to most computing science researchers. Nevertheless such a structured approach to human evaluation experiments is rarely followed in research. That the methodology proposed here is so simple means it should be straightforward to implement for most experiments. Doing so comes with the benefits of reduced manual data entry errors, improved repeatability, ease of pre-registration, and assurance to readers that the experiment has not undergone ad hoc and biased changes as the researcher made observations during the process.

[5] This method is inspired by that of Hosking and Lapata (2021); Hosking et al. (2022).

Acknowledgments

We would like to thank the reviewers for their helpful and insightful feedback. Both authors benefit from being members of the SFI Ireland funded ADAPT Research Centre.

References

Mohammad Arvan and Natalie Parde. 2023. Human evaluation reproduction report for data-to-text generation with macro planning. In *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*, pages 89–96, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Kathy Barker. 1998. *At the bench a laboratory navigator*, first edition. Cold Spring Harbor Laboratory Press.

Anya Belz, Simon Mille, and David M. Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 183–194, Dublin, Ireland. Association for Computational Linguistics.

Anya Belz, Simon Mille, João Sedoc, Craig Thomson, and Rudali Huidrom. 2024. Tutorial on human evaluation of nlp system quality at inlg'24. In *Proceedings of the 17th International Conference on Natural Language Generation: Tutorial Abstracts*, Tokyo, Japan.

Anya Belz and Craig Thomson. 2024. The 2024 repronlp shared task on reproducibility of evaluations in nlp: Overview and results. In *Proceedings of the 4th Workshop on Human Evaluation of NLP Systems*.

Anya Belz, Craig Thomson, Ehud Reiter, Gavin Abercrombie, Jose M. Alonso-Moral, Mohammad Arvan, Anouck Braggaar, Mark Cieliebak, Elizabeth Clark, Kees van Deemter, Tanvi Dinkar, Ondřej Dušek, Steffen Eger, Qixiang Fang, Mingqi Gao, Albert Gatt, Dimitra Gkatzia, Javier González-Corbelle, Dirk Hovy, Manuela Hürlimann, Takumi Ito, John D. Kelleher, Filip Klubicka, Emiel Krahmer, Huiyuan Lai, Chris van der Lee, Yiru Li, Saad Mahamood, Margot Mieskes, Emiel van Miltenburg, Pablo Mosteiro, Malvina Nissim, Natalie Parde, Ondřej Plátek, Verena Rieser, Jie Ruan, Joel Tetreault, Antonio Toral, Xiaojun Wan, Leo Wanner, Lewis Watson, and Diyi Yang. 2023a. Missing information, unresponsive authors, experimental flaws: The impossibility of assessing the reproducibility of previous human evaluations in NLP. In *Proceedings of the Fourth Workshop on Insights from Negative Results in NLP*, pages 1–10, Dubrovnik, Croatia. Association for Computational Linguistics.

Anya Belz, Craig Thomson, Ehud Reiter, and Simon Mille. 2023b. Non-repeatable experiments and non-reproducible results: The reproducibility crisis in human evaluation in NLP. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3676–3687, Toronto, Canada. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

C. Gaspar, F. Alessio, L. Cardoso, M. Frank, Garnier J.C., E. v. Herwijnen, R. Jacobsson, B. Jost, N. Neufeld, R. Schwemmer, O. Callot, and B. Franek. 2021. The lhcb experiment control system: On the path to full automation. In *13th International Conference on Accelerator and Large Experimental Physics Control Systems*, pages 20–23.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *J. Artif. Int. Res.*, 77.

Matthew Gentzkow and Jesse M. Shapiro. 2014. *Code and Data for the Social Sciences: A Practitioner's Guide*. University of Chicago mimeo.

Javier González Corbelle, Jose Alonso, and Alberto Bugarín-Diz. 2023. Some lessons learned reproducing human evaluation of a data-to-text system. In *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*, pages 49–68, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Tom Hosking and Mirella Lapata. 2021. Factorising meaning and form for intent-preserving paraphrasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1418, Online. Association for Computational Linguistics.

Tom Hosking, Hao Tang, and Mirella Lapata. 2022. Hierarchical sketch induction for paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2489–2501, Dublin, Ireland. Association for Computational Linguistics.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2024. Analyzing Dataset Annotation Quality Management in the Wild. *Computational Linguistics*, pages 1–50.

Klaus Krippendorff. 2004. Reliability in content analysis. *Human Communication Research*, 30(3):411–433.

William H. Kruskal and W. Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621.

Yiru Li, Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2023. Same trends, different answers: Insights from a replication study of human plausibility judgments on narrative continuations. In *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*, pages 190–203, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Yvette Oortwijn, Thijs Ossenkoppele, and Arianna Betti. 2021. Interrater disagreement resolution: A systematic procedure to reach consensus in annotation tasks. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 131–141, Online. Association for Computational Linguistics.

David Robertson, Siu-wai Leung, and Dietlind Gerloff. 2009. Welcome to automated experimentation: A new open access journal. *Automated experimentation*, 1(1):1–2.

Jie Ruan, Wenqing Wang, and Xiaojun Wan. 2024. Defining and detecting vulnerability in human evaluation guidelines: A preliminary study towards reliable NLG evaluation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7965–7989, Mexico City, Mexico. Association for Computational Linguistics.

S. S. Shapiro and M. B. Wilk. 1965. An analysis of variance test for normality (complete samples)†. *Biometrika*, 52(3-4):591–611.

Anastasia Shimorina and Anya Belz. 2022. The human evaluation datasheet: A template for recording details of human evaluation experiments in NLP. In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 54–75, Dublin, Ireland. Association for Computational Linguistics.

Craig Thomson, Ehud Reiter, and Belz Anya. 2024. Common flaws in running human evaluation experiments in nlp. *Computational Linguistics*.

Emiel van Miltenburg, Anouck Braggaar, Nadine Braun, Debby Damen, Martijn Goudbeek, Chris van der Lee, Frédéric Tomas, and Emiel Krahmer. 2023. How reproducible is best-worst scaling for human evaluation? a reproduction of 'data-to-text generation with macro planning'. In *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*, pages 75–88, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Lewis Watson and Dimitra Gkatzia. 2023. Unveiling NLG human-evaluation reproducibility: Lessons learned and key insights from participating in the ReproNLP challenge. In *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*, pages 69–74, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Lewis N. Watson and Dimitra Gkatzia. 2024. ReproHum #0712-01: Reproducing human evaluation of meaning preservation in paraphrase generation. In *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval) @ LREC-COLING 2024*, pages 221–228, Torino, Italia. ELRA and ICCL.

A A note on Amazon MTurk

We designed this experiment for Amazon Mechanical Turk in order to make our examples clearer; many researchers will be familiar with MTurk. However, there is a problem with our design in that MTurk (using the web interface) does not prevent workers from accepting multiple lists. In practive, we suggest the use of Prolific, using an integration such as the code from Watson and Gkatzia (2024) to ensure that each participant is allocated only one list. [6][7]

B Question and interface design

The design of the interface and the wording of the question that participants are asked is vitally important in any human evaluation (Howcroft et al., 2020; Belz et al., 2020). However, these issues are not the focus of this paper. If there is anything wrong with the process, question, or interface, they will be recorded as such. The crucial thing in terms of the repeatability of the experiment is that they are recorded.

An HTML file in Mechanical Turk format has been included with supplementary material. However, since the focus of this paper is on recording the process of the experiment, the question, interface, and indeed the system output texts are just placeholders.

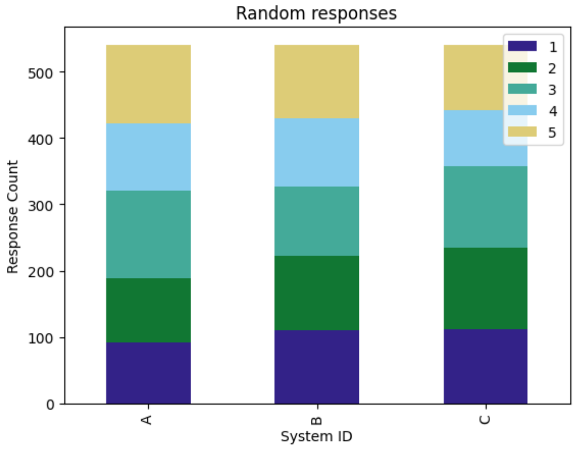

Figure 2: Bar chart showing the distribution of responses in the dummy results when responses are allocated randomly.

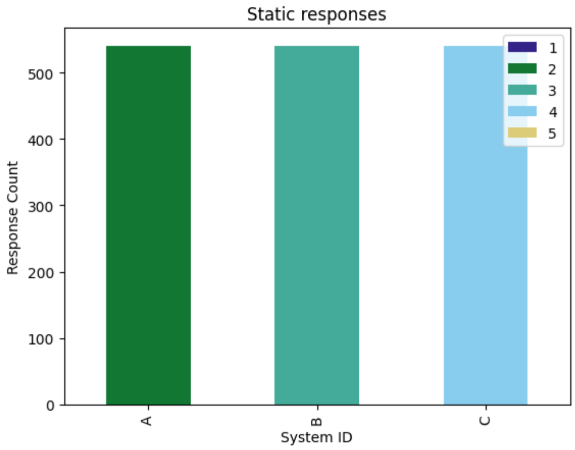

Figure 3: Bar chart showing the distribution of responses in the dummy results when each system is always assigned the same score.

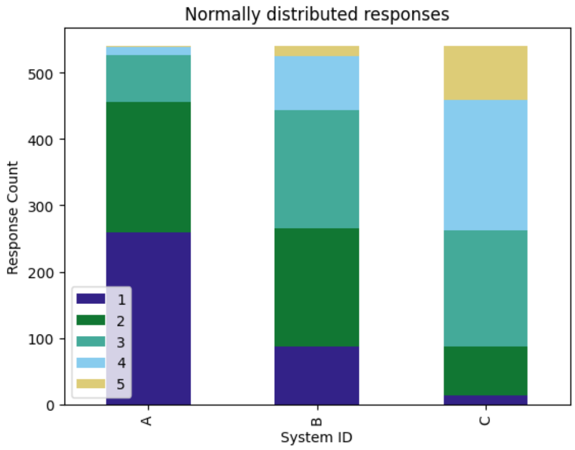

Figure 4: Bar chart showing the distribution of responses in the dummy results when generated as normal distributions about a mean.

[6] https://www.mturk.com

[7] hrttps://prolific.com