The INLG 2024 Tutorial on Human Evaluation of NLP System Quality: Background, Overall Aims, and Summaries of Taught Units

Anya Belz[1], João Sedoc[2], Craig Thomson[1], Simon Mille[1], Rudali Huidrom[1]

[1]ADAPT, Dublin City University, [2]New York University

Abstract

Following numerous calls (e.g. van der Lee et al., 2019; Howcroft et al., 2020; Thomson et al., 2024) in the literature for improved practices and standardisation in human evaluation in Natural Language Processing over the past ten years, we held a tutorial on the topic at the 2024 INLG Conference. The tutorial addressed the structure, development, design, implementation, execution and analysis of human evaluations of NLP system quality. Hands-on practical sessions were run, designed to facilitate assimilation of the material presented. Slides, lecture recordings, code and data have been made available on GitHub.[1] In this paper, we provide summaries of the content of the eight units of the tutorial, alongside its research context and aims.

1 Research Context and Aims

Human evaluation is widely considered the most reliable form of evaluation in Natural Language Processing (NLP), but recent research has thrown up a number of concerning issues, including in the design (Belz et al., 2020; Howcroft et al., 2020) and execution (Thomson et al., 2024) of human evaluation experiments, but also obstacles in adopting good practices (Gehrmann et al., 2023). Standardisation and comparability across different experiments is low, as is reproducibility in the sense that repeat runs of the same evaluation often do not support the same main conclusions, quite apart from not producing similar scores. The situation is likely to be in part due to how human evaluation is viewed in NLP: not as something that needs to be studied and learnt before venturing into conducting an evaluation experiment, but something that anyone can throw together without prior knowledge by pulling in a couple of students from the lab next door.

Our aim with this eight-unit tutorial is primarily to inform participants about the range of options available and choices that need to be made when creating human evaluation experiments, and what the implications of different decisions are. Moreover, we present best practice principles and practical tools that help researchers design scientifically rigorous, informative and reliable experiments. The tutorial is structured into seven units each consisting of a lecture and (in the case of Units 3, 4, 5, and 6) a brief hands-on exercise, followed by an extended practical session (Unit 8), where participants create evaluation experiments and analyse results from them, using tools and other resources provided as part of the tutorial.

We aim to address all aspects of human evaluation of system outputs in a research setting, equipping participants with the knowledge, tools, resources and hands-on experience needed to design and execute rigorous and reliable human evaluation experiments. Publicly shared materials and online resources will continue to support participants in developing and conducting experiments after the tutorial.[1]

2 Tutorial Unit Summaries

Unit 1: Introduction

In Unit 1 our aims are (i) to give a first idea of what human evaluation means in NLP; (ii) to summarise the current state of human evaluation in NLP; and (iii) to survey some of the challenges and issues that have been identified, and how current research is beginning to address them.

Unit 1 lays the groundwork for the tutorial, starting with an example from image caption generation, shown in Figure 1. Among the possible captions for the labelled image are:

- *Dining table with breakfast items*
- *Dining table with breakfast items, including*

[1] https://github.com/Human-Evaluation-Tutorial/INLG-2024-Tutorial

Proceedings of the 17th International Natural Language Generation Conference: Tutorial Abstract, pages 1–12

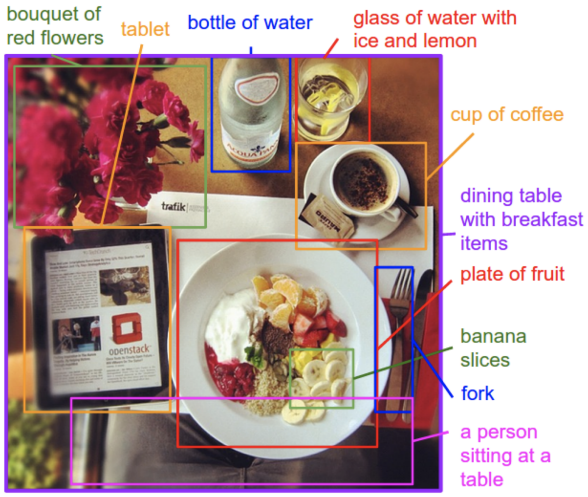

Figure 1: Example of a labelled image from Karpathy and Fei-Fei (2015) as input to caption generation.

> *bouquet of red flowers, tablet, bottle of water, glass of water with ice and lemon, cup of coffee, plate of fruit, banana slices, fork, and a person sitting at a table*

- *My dream breakfast*
- *Where's the bacon and eggs?!*

So how do we decide which of these captions is a good one for the image, and/or which is better than others? NLP uses a range of different ways of answering questions about system quality including metric-based, human-evaluated, and most recently approaches known (self-explanatorily) by the term LLM-as-judge. Examples of human-evaluated approaches are asking some humans (i) to rate the quality of each output in terms of a given criterion (*Fluency*, *Grammaticality*, *Input Coverage*, etc.); (ii) to perform a task with/without the outputs, and measuring relative performance (number of post-edits, speed of finding searched-for items, etc.); (iii) to interact with the system, and taking automatically computable measurements during the interaction (task completion, click rates, reaction time, etc.); and (iv) to interact with the system, followed by questions about their experience (overall satisfaction, ease of use, understandability, etc.).

Different evaluation methods can yield different estimates of system quality and system rankings. In selecting evaluation methods, aspects to take into account include the application context (in a social media context, the last two captions above may be most suitable), and user-specific characteristics (deciding between the last two captions above depends on the user's taste in breakfast).

The field of NLP has a 40+ year history of conducting human evaluation experiments to determine system quality, but very few established shared standards and methods for human evaluation. As early as the 1980s, Spärck Jones (1981) advocated the systematic testing of variations of data sets and systems in "a uniform framework for system characterisation and evaluation." For human evaluation at least, we are still far from such a framework, and this is likely a substantial contributing factor to a range of issues and challenges in human evaluation of NLP systems that have been identified recently:

1. Lack of standardisation in what is being evaluated: does one evaluation of 'Fluency' assess the same thing as another? (Howcroft et al., 2020).
2. Low levels of reproducibility to the point where same main conclusions are often not supported by otherwise identical human evaluations (Belz and Thomson, 2024; Thomson et al., 2024).
3. Poor practice in designing and executing human evaluation experiments, e.g. bugs, reporting errors, ad hoc interference in live experiments, etc. (Thomson et al., 2024).
4. Loose application of experimental and statistical methods and principles, e.g. not testing assumptions, unsuitable significance tests, over-reliance on post-hoc testing.

The tutorial aims to contribute to addressing these issues through providing structured, step-by-step information and guidance on how to put together scientifically rigorous experiments that produce reliable answers to questions like the above (Which of the captions are good? Better? More correct? On what grounds?).

Unit 2: Development and Components of Human Evaluations

The aims in Unit 2 are (i) to introduce core standard terminology for human evaluation in NLP; (ii) to examine the components and processes common to all human evaluations, introducing a standard framework comprising (a) a standard process diagram for human evaluations, and (b) a standard four-phase decomposition of the steps in creating and running a human evaluation.

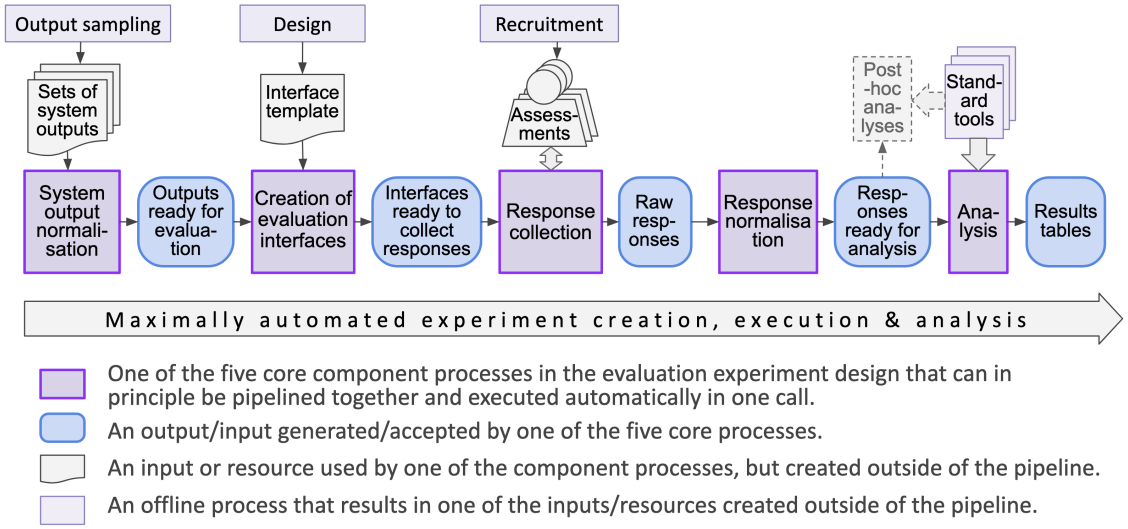

Figure 2: Generic process diagram of human evaluations showing the five core processes (output normalisation, interface instantiation, response collection, response normalisation, and analysis), and three offline processes.

Figure 2 shows the five core processes and three secondary processes in a human evaluation experiment: (i) system output normalisation; (ii) evaluation interface instantiation; (iii) response collection; (iv) response normalisation; and (v) aggregation and analysis of results. Ideally, the whole experiment pipeline is implemented in such a way that the five core processes can be run as a single pipeline that maps system outputs to results tables, as automation reduces human error (Thomson et al., 2024; Thomson and Belz, 2024).

All eight processes are specified in Phase I of experiment development (Design, covered in Units 3 and 4), implemented as code or at least as a formal process protocol in Phase II (Implementation, Unit 6), and executed in Phase III (Execution, Unit 7). Results from the execution are analysed in Phase IV (Analysis, Unit 5). Figure 3 shows the four phases and the resources resulting from each.

The main tasks in Phase I are (i) formulation of research question(s) and hypotheses, including selection of systems and power calculations; (ii) selection of quality criteria and evaluation modes; (iii) selection of methods for system output sampling and output normalisation; (iv) specification of experiment design properties such as the evaluation interface template, rating instrument and response values, evaluator recruitment and training, etc.; (v) specification of evaluation interface instantiation and response collection processes; (vi) selection of methods for evaluator recruitment and training; (vii) specification of methods for normalisation, aggregation and analysis of responses; (viii) review of design in terms of ethical considerations; and (ix) completing a human evaluation datasheet (HEDS) (Shimorina and Belz, 2022).

In Phase II the experiment design from Phase I is implemented as code or written processes. Each of the core processes defined in Phase 1 should be implemented, ideally as code scripts, but otherwise they should be written down as clear step by step instructions that any researcher could follow in order to execute the experiment. As many of these core processes as possible should pipelined together, e.g., with one pipeline running everything before Response Collection and another pipeline running everything after, and manual steps defined for the participant to collect responses. During the course of implementation, it may be necessary to return to and update the design, such that the implementation never deviates from it. In developing code, good coding practices need to be applied just as in system development, including code testing, review, and documentation. If any changes have been made, the HEDS sheet needs to be updated.

Phase III is the execution of the response collection code or protocol. This will happen any number of times during testing (pre-final execution), after which the experiment itself is run as the final execution. Pre-final execution iterates as needed over tests including interface robustness testing and a pilot test, normally with a smaller number of evaluators and evaluation items than planned for the actual, final run of the experiment. After this, pilot responses are tested for inter and intra-annotator agreement, feedback is collected from pilot eval-

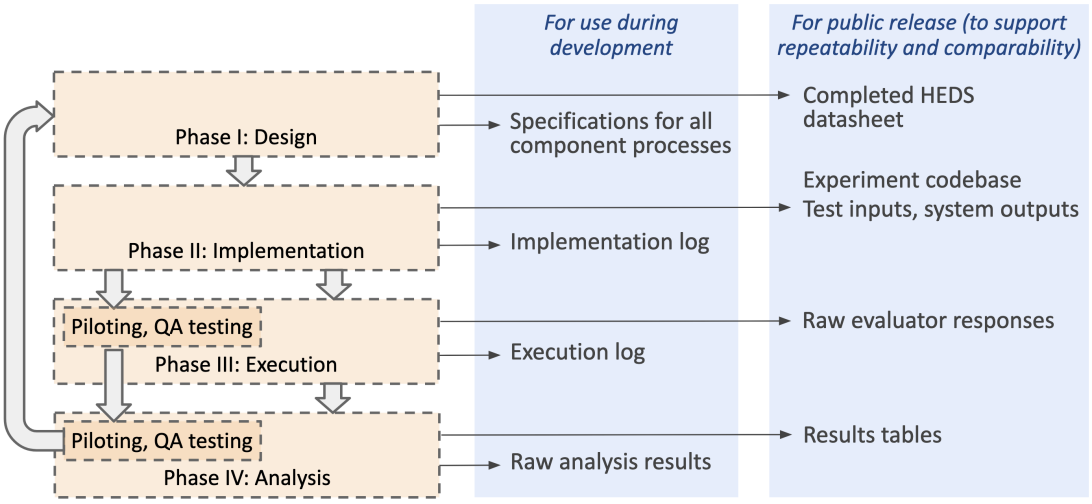

Figure 3: The four phases in creating and running a human evaluation; resources resulting from each phase.

uators regarding understandability and task complexity, improvements are collated, updated in the design, and then implemented, and the HEDS sheet is updated one final time and used in preregistration. During the final execution, response collection is run with the final number of evaluators/items.

Finally, in Phase IV, the response aggregation and analysis are executed exactly as preregistered. If needed, additional posthoc tests can be run, including multiple test corrections as needed. New scripts may need to be created to generate any additional results tables from posthoc tests. Results should be reported in two separate parts, always clearly stating which each is: results from preregistered tests or results from post-hoc tests.

Unit 3: Quality Criteria and Evaluation Modes

In Unit 3 our aims are (i) to introduce the concept of null hypothesis testing and relate it to the formulation of research questions; (ii) to deepen understanding of the concepts of quality criteria and evaluation modes first introduced in Unit 2, and of their role in formulating research questions; (iii) to introduce a taxonomy of quality criteria and the QCET tool (Belz et al., 2024) for interacting with it, which facilitate designing standardised hence comparable evaluation measures; and (iv) to explain the connection between evaluation measures and the formulation of research questions, as separate from specifying experiment properties.

Unit 3 takes the first step in Phase I (Design): formulating the research question(s) and corresponding hypotheses which necessarily includes specifying quality criterion (Belz et al., 2024) and evaluation modes (Belz et al., 2020).

Suppose we have created a new language generation system M_{new} and want to know if it performs better than an existing system M_{old}. One of the things we need to decide is what we mean by 'better than.' Suppose we are interested in improving grammaticality of outputs and wish to assess if M_{new} is better in this respect. The evaluation criterion *Grammaticality* assesses the **correctness** of the **form** of outputs **in their own right**, using property values from the quality criterion taxonomy we introduce in this unit (see below). E.g. we know the following sentences are grammatically correct without considering their meaning, or anything other than the sentence itself:

> *Colorless green ideas sleep furiously.*
> *All mimsy were the borogoves, And the mome raths outgrabe.*

For a fully specified research question, we also need to decide evaluation modes, as the answer may be different depending on which modes we choose: Do we just want to know whether M_{new} outputs are more grammatical than M_{old} outputs (relative mode), or also quantify by how much (absolute mode)? Are we interested more in users' perception of the system's grammaticality (subjective mode), or measuring the degree to which its outputs conform to a given notion of grammar (objective mode)? Do we want to assess outputs directly (intrinsic mode), or in terms of their effect on something external to the system, e.g. how many post-edits a user performs (extrinsic mode)?

Suppose we decide to assess our quality criterion

Grammaticality in *absolute*, *subjective* and *intrinsic* evaluation modes (by far the most common combination of evaluation modes in NLP). This gives us *Absolute, Subjective, Intrinsic Grammaticality* as the evaluation measure m. This could, at a later stage in the Experiment Design phase, be decided to be assessed by asking evaluators to rate each system output individually on a scale of 1–5 (there are many other options). But that is part of how we choose to find an answer for our research question (Experiment Design, Unit 4), whereas the evaluation measure (quality criterion + evaluation modes) is part of the research question itself.

Next we need to formulate the research question; two common forms are:

A. Is M_{new} more absolutely, subjectively and intrinsically grammatical than M_{old} ?

B. Which of M_{new} and M_{old} is more absolutely, subjectively and intrinsically grammatical?

The corresponding hypotheses that are tested by the evaluation experiment are then:

A. ***Null hypothesis***: There is no difference between M_{new} and M_{old} in terms of absolute, subjective and intrinsic grammaticality.
Alternative hypothesis: M_{new} is more absolutely, subjectively and intrinsically grammatical than M_{old} .

B. ***Null hypothesis***: There is no difference between M_{new} and M_{old} in terms of absolute, subjective and intrinsic grammaticality.
Alternative hypotheses:
M_{new} is more grammatical in absolute, subjective and intrinsic terms than M_{old} .
M_{old} is more grammatical in absolute, subjective and intrinsic terms than M_{new} .

The choice of research question impacts the statistical power of the experiment and the types of statistical tests that can be applied – we will come back to this in Unit 5. NB: Answering research questions of type A can never produce evidence that M_{old} is better, only either that M_{new} is better or that M_{new} is not better. In contrast, answering research questions of type B can produce evidence either that M_{old} is better, or that M_{new} is better (or alternatively, that no evidence is found supporting either conclusion).

Once we have chosen **quality criterion** and **evaluation modes** we have a fully specified **evaluation measure** to incorporate in our chosen research question (as above). We still need to specify the experiment properties (Unit 4) for a fully specified **evaluation method**. The relationships between these four elements can be summarised as follows:

- Quality criterion + evaluation mode = evaluation measure;
- Evaluation measure + experimental design = evaluation method.

We first summarise quality criteria, then evaluation modes. We use the QCET taxonomy tool (Belz et al., 2024) which is an extension of the 71 standardised quality criteria (QCs) and taxonomy from Howcroft et al. (2020), and facilitates perusal via an interactive user interface. Recall the example from earlier: Grammaticality assesses the *correctness* of the *form* of outputs *in their own right*. The terms in italics refer to the main three levels, or ***QC properties***, in the taxonomy where nodes branch along three dimensions (for full details see Howcroft et al. (2020) and Belz et al. (2020)):

i. **Type of quality** assessed: *Correctness, Goodness, Feature*;

ii. **Aspect of outputs** assessed: *Form, Content, Both form and content*; and

iii. **Frame of reference** relative to which system quality is assessed: *Outputs In their own right, Relative to the inputs, Relative to an external frame of reference*.

Evaluation modes (Belz et al., 2020) are orthogonal to quality criteria, i.e. any given quality criterion can be combined with any of the modes:

i. **Objective vs. subjective**: Examples of objective assessment include any automatically counted or otherwise quantified measurements such as mouse-clicks, occurrences in text, etc. Subjective assessments involve ratings, opinions and preferences by evaluators.

ii. **Absolute vs. relative**: whether evaluators are shown outputs from a single system during evaluation (absolute), or from multiple systems in parallel (relative), in the latter case typically ranking or preference-judging them.

iii. **Extrinsic vs. intrinsic**: in extrinsic evaluation, system performance is assessed in terms of the system's effect on something external to the

system, e.g. how it affects the performance of an embedding system or of a user at a task; in intrinsic evaluation, outputs are assessed only within the system context (can include relative to inputs or to an expected standard).

Unit 4: Experiment Design

In Unit 4 we (i) take a closer look at the remaining steps involved in Phase I of human evaluation development (Experiment Design); (ii) present a range of representative design options for each step and consider their suitability in different evaluation contexts; and (iii) introduce the Human Evaluation Data Sheet (HEDS) for capturing details of an experiment and for use in preregistration (Shimorina and Belz, 2022).

Unit 3 got us as far as the evaluation measure m comprising a quality criterion and three evaluation modes. In this unit, we look at those aspects of experimental design that provide the remaining elements for a fully specified evaluation method E_m that can be used to obtain measured values v_i for each system M_i (represented by its outputs o_i^s for a given test set s), and evaluation measure m, or:

$$E_m : (o_i^s, s) \mapsto v_i \quad (1)$$

We can think of m as *what* is being evaluated (part of the research question). Experiment design specifies *how* to evaluate m (how to answer the research question). Experiment design can be broken down into different aspects which we call **experiment design properties**; these are covered in HEDS Section 3 and Questions 4.3.1–4.3.9. In the tutorial, we briefly go through all 25 properties that need to be specified, and then go into the following property sets in more detail:

- Rating instrument, response collection method and basic evaluator interface.
- Methods for postprocessing, aggregation and analysis of results.
- Evaluator recruitment, training and monitoring.
- Review of design in terms of ethical considerations.

Some of the choices for rating instrument are: Numerical Rating Scale, Slider Scale, Verbal Descriptor Scale, Likert Scale (agreement with statement), Rank Ordering, Text Annotation, Post-editing, Free-text Entry, and Item Counting. The first five of these need the numerical ranges associated with them specified. In all cases, visual appearance and labels also need to be decided.

Response collection (or elicitation) can be done in a number of different ways, including: (Dis)agreement with quality statement, Direct quality estimation, Relative quality estimation, Qualitative feedback, Evaluation through post-editing/annotation, User-text interaction measurements, Task performance measurements, and User-system interaction measurements. These are about how evaluators interact with the rating instruments above during assessments. Note that the two properties are not orthogonal, as some combinations are impossible. E.g. you can't select from a rating scale via text annotation.

When selecting rating instrument and response collection methods, care needs to be taken to keep the cognitive load on evaluators to a minimum, as intra- and inter-annotator agreement tends be lower, and errors more frequent, with higher cognitive load. For example, asking evaluators to manually count errors in a stretch of text with several sentences, and then to enter a comma-separated list with one count for each sentence imposes a very high cognitive load on evaluators.

Evaluator recruitment deserves more attention than it sometimes gets (many papers do not even mention how this was done). We need to decide aspects such as how many evaluators we need, what their qualifying characteristics should be, what training and practice they do, and what information you gather about them as part of the experiment.

Instructions and examples given to participants should be designed such that they are clear, concise, and free of unnecessary technical jargon. The process by which participants are trained to perform the evaluation also needs to be specified, for example by planning a training session with the same level of detail that one might plan a teaching session for students. The conditions under which participants will be excluded, and how replacement judgments will be obtained, must also be decided, e.g., by defining attention checks.

All experiments should be performed in an ethical manner and this is usually ensured by submitting the experiment design for review to a research ethics committee (REC). We provide an overview of the questions that are likely to be asked during such a review, as well as actions that can be taken to ensure experiments adhere to ethical standards. For researchers without access to an REC, we reference resources such as the guidelines provided by UK

funding body The Economic and Social Research Council.[2]

In the tutorial materials, we go through all the above properties and more in detail. Once every aspect of the experiment has been fully specified, the HEDS datasheet can be completed. The next step in developing the experiment is then Implementation (Phase II) which we cover in Unit 6.

Unit 5: Statistical Analysis of Results

The aims of Unit 5 are (i) to discuss data transformations and model-free evidence; (ii) to revisit the analysis of results from a statistical perspective; (iii) to present null hypothesis significance testing, statistical significance tests, and power analysis; (iv) to gain an understanding of pre-registration and confirmatory vs. exploratory hypothesis testing; (v) to understand annotator agreement metrics; and (vi) to understand post-hoc analysis and multiple hypothesis testing corrections for false discovery. The unit also covers essential concepts such as Type I and Type II errors, significance level, and power. Aside from the previous units, the only assumed knowledge is a background on probability.[3]

Rigorous statistical analysis of results requires one to be familiar with concepts of simple data analysis and transformations, data reliability analysis, exploratory data analysis, and hypothesis testing (especially in the context of model reasoning[4] and outcome reasoning[5]) (Rodu and Baiocchi, 2023).

Simple data analysis and transformations are fundamental steps to any analysis of results. Indeed even in the pilot phase of an experiment, one may decide to change the experiment as a result of this analysis. We discuss how a set of typical analyses examines effect sizes, outliers, normality tests, scale ranges, and more, and consider how data transformations are decided.

Next, we start with model-free evidence (Chatfield, 1985; Tufte and Graves-Morris, 1983), emphasising the importance of examining trends and patterns directly from the data without imposing a specific theoretical model. We introduce participants box-plot visualisations to modern visualisations, like the q-q box plot (Rodu and Kafadar, 2022). This is followed by taking the WebNLG 2017 (Shimorina et al., 2018) human evaluation data and showing a cumulative density plot on superiority (see Figure 4).

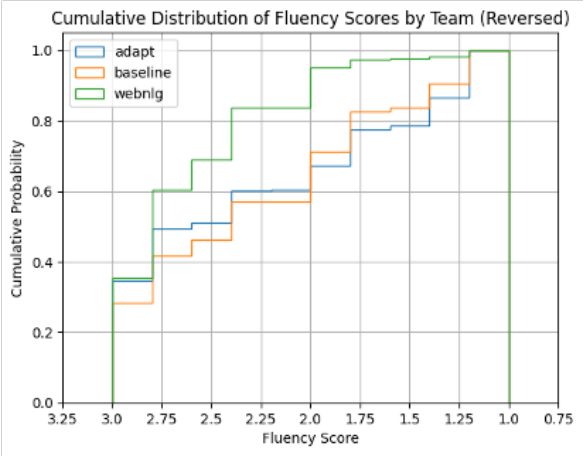

Figure 4: Model-free cumulative density plot where more area under the curve shows superiority (3 is the highest rating and 1 is lowest).

A main focus in the unit is a discussion of null hypothesis significance testing (NSHT). We start by presenting the common assumptions behind the NSHT methodology. Next, we focus on three common scenarios that NLP researchers face. The underlying test statistic for significance testing is explained and a flowchart is provided that serves as a guide for selection.

The unit continues by making connections between NSHT and pre-registration (a formalisation of NSHT). Pre-registration, the practice of specifying hypotheses and analysis plans before conducting a study, is crucial for avoiding biases and ensuring the validity of research findings. We discuss power analysis and sample size requirements.

Aside from confirmatory analysis using NSHT, we also discuss exploratory analysis, multiple hypothesis test correction, and post-hoc tests.

Next, we discuss annotation reliability and the most common measures used as well as further considerations. Specifically, we cover Cohen's Kappa and Krippendorff's alpha as well as other measures. We discuss what acceptable ranges mean and why. These are fundamental components of both the pilot phase as well as the overall evaluation process.

Finally, the unit concludes with a practical code-based session, allowing participants to apply the statistical concepts and techniques learned throughout the unit. This hands-on experience reinforces their understanding and equips them with the skills necessary to conduct rigorous statistical analyses

[2] ukri.org/councils/esrc/guidance-for-applicants/research-ethics-guidance/

[3] We provide pointers as well as the necessary materials.

[4] This is testing a novel change to existing practices (e.g., a new type of model).

[5] This is comparing systems (e.g., system ranking).

in their own research.

Due to the time limitations of this tutorial we are not able to cover all of the possible material. We advise participants on *essential readings* and *further readings*, including the following.

The essential readings for statistical inference in null hypothesis testing start with a guide of statistical tests (Dror et al., 2018) and continues with a must-read on power analysis from Card et al. (2020). Next, we continue with annotator agreement and reliability measure and discussion and highlight the work of Artstein (2017) and Rottger et al. (2022).

For further readings, we first point the participants to the following textbooks: (i) a specialized book for NLP "Statistical Significance Testing for Natural Language Processing" (Dror et al., 2020), (ii) a standard textbook for statistical inference for non-statisticians by "Mathematical Statistics: Basic Ideas and Selected Topics" (Bickel and Doksum, 2015), and (iii) a more complete and rigorous reading "All of Statistics: A Concise Course in Statistical Inference" (Wasserman, 2013).

Finally, for extended-depth readings, we split these into (i) depth in hypothesis testing and its alternatives, and (ii) annotator agreement assessment. The highlighted readings into modern views of hypothesis testing and its pitfalls are discussions of choices of significance thresholds (and associated p-values).

Unit 6: Experiment Implementation

The aims of Unit 6 are to (i) give an overview of how the experiment design from Units 3 and 4 is implemented; (ii) cover each component process from Figure 2 in turn, describing the inputs, outputs, and functionality of the process, with examples; (iii) show how the component processes can be pipelined in order to maximise automation; (iv) discuss updating the design, based on issues raised during implementation; and (v) conclude with an overview of good coding practices.

Each component shown in Figure 2 incorporates some of the design specified in Phase I. The core five processes (paragraph headings in **bold**) and three offline processes (paragraph headings in *italics*) are as follows:

Output sampling: Once the systems to be evaluated have been selected, samples of their behaviour need to be obtained for presentation to evaluators. This usually means generating system outputs for the same set of inputs, but sometimes is a sequence of user and system turns (as in dialogue tasks), or other user-system interactions. Inputs need to be selected so their characteristics are representative of the system task as a whole, e.g. by stratified random sampling. A large enough sample needs to be obtained for the desired statistical power of the experiment (see Unit 5).

System output normalisation: Outputs from different systems may have different tokenisation, capitalisation, etc. These outputs need to be normalised such that any differences in syntax or formatting are removed as to not affect participant judgment. A script is required to perform this normalisation is applied without any human errors. Occasionally, this kind of post-processing is part of the task (as in WebNLG 2023) in which case this step is omitted.

Interface template design: The interface template that was designed in Phase I (Unit 4) must be created, a.g. as an HTML form or a spreadsheet. Simple survey tools such as Google Forms or Microsoft Forms can be used, although limits to their functionality affect available options for the experiment design.[6,7] Crowd platforms can also be used, e.g., Amazon Mechanical Turk (MTurk) generally requires that an HTML template, with special markup indicating slot variables, be populated by a CSV file where each row contains the item(s) one participant must rate as part of the Human Intelligence Task (HIT).[8] Other platforms, such as Prolific, allow for integration with more flexible form builders such as Qualtrics, or even custom web servers such as that of Watson and Gkatzia (2024).[9,10]

Generation of evaluation interfaces: In this core process, the interface template is populated with appropriate evaluation items for each participant. This must be implemented as a code process to avoid human error. In the case of crowd platforms, evaluation interfaces might not be generated in advance of running the experiment, but rather at execution time (still using the same interface template and data file containing evaluation items). For example, when running a project on MTurk, a CSV containing evaluation items for all participants is

[6] google.com/forms/about
[7] forms.office.com
[8] mturk.com
[9] prolific.com
[10] qualtrics.com

uploaded when the experiment execution is commenced, with the MTurk server then generating the interfaces that are shown to participants at the time of each participant previewing or accepting the work. The end result is the same, participants are shown evaluation interfaces containing their evaluation items, the change is only in the time at which the predefined evaluation interface generation process is executed.

Recruitment of participants: Following the design specifications, the specific type of evaluator required needs to be recruited, and provided with instructions, training, example evaluation exercises and opportunities to ask questions. A qualification exercise is also advisable to ensure participants can perform the task required of them. Fair payment should also be ensured, e.g. following the ReproHum Project guidance (Belz and Thomson, 2023).

Response collection: A process should be implemented for giving participants access to evaluation interfaces, as well as for the collection of completed interfaces and the raw responses they contain. The amount of manual work by the researcher should be kept at a minimum for this process, and all actions they will take documented in advance. For example, if 30 evaluation interfaces in the form of spreadsheets (hosted on the cloud as Google Sheets) are to be distributed to 30 participants by email, it would be best to create a distribution spreadsheet that links the spreadsheet URLs for each participant to their email, and then perform a mail merge to send the details to each participant. Automated tests can also be created to, for example, check that each spreadsheet can only be accessed by 1 email account, and that this account matches the email in the distribution spreadsheet.[11]

Response normalisation: The raw responses output by the previous component process are not normally in a format that is suitable for statistical analysis. E.g., participants may have completed a spreadsheet such as that in Figure 5, where responses are entered on every other row following an initial offset. This core process extracts raw responses from the evaluation interfaces, in a structured data format that is suitable for analysis, and includes all relevant identifiers such as input data ids, system ids, and anonymised participant ids.

[11]This can be done using the Google Drive API https://developers.google.com/drive/api/guides/about-sdk

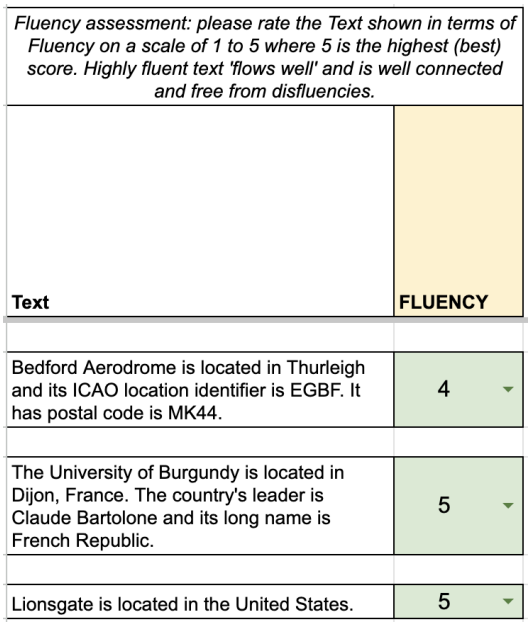

Figure 5: Example interface from the WebNLG 2023 human evaluation (Cripwell et al., 2023).

Analysis: Our primary analysis should be encoded in advance such that it can simply be run at the time of experiment execution, using the postproccessed responses, and with no intervention from the researcher. It is possible to encode logical conditions, for example we may run a test for data normality and *if* the data is normal *then* we will run a predefined parametric test such as an ANOVA, *else* we will run a nonparametric test such as Kruskal-Wallis.

Running examples are included for each component process, with one of these forming the basis of a short practical session where there is a high-level code overview of an experiment design implemented in python. For each component process, participants are shown the functions that were created, as well as the shape of the input/output data.

Finally, this unit covers good coding practices, with a focus on Python code (as is common in NLP and Machine Learning experiments). Attendees are shown ways in which they can keep their projects organised, using Cookie Cutter templates as a starting point for projects that share a commonality (such as human evaluation).[12] The use of linters and formatters is discussed, as well as testing, documentation, and code review. Simple methods of writing clearer code are discussed, with attendees directed to resources such as the Mineault and Community (2021) for further reading.

[12]https://github.com/cookiecutter/cookiecutter

Unit 7: Experiment Execution

The aims of Unit 7 are to (i) describe execution for testing purposes (called pre-final execution below), including interface testing and piloting; (ii) describe preregistration, keeping a record of the experiment execution, and documenting any unavoidable changes that occur during experiment execution; (iii) discuss the process of recruiting participants, with a particular focus on ethical issues and fair treatment of participants.

Since the experiment design (Phase I, Units 3 and 4) has been fully implemented (Phase II, Unit 6) in advance, the execution of the experiment itself for the purpose of collecting responses should be straightforward: written procedures are followed and code pipelines are run. However, there are still some additional issues that we should be aware of when executing our experiments.

We look at three different purposes for which experiments are executed: (i) testing (pre-final execution); (ii) running the actual experiment as it will be reported (final execution); and (iii) reproducibility testing (post-final execution). When discussing pre-final execution, we look at interface testing, pilot experiments (typically with smaller numbers of participants and evaluation items), checks for intra- and inter-annotator agreement, obtaining qualitative feedback from pilot participants, and finally updating the experiment design (by going back to previous phases if necessary) based on issues raised during pre-final execution.

We then discuss the final execution of the experiment, which starts with creating the preregistration, beyond which point the design and implementation become fixed. Pre-registration sites such as aspredicted.org can be used to create private, timestamped preregistrations that can be made public upon completion of the experiment. GitHub repositories similarly have timestamps for commits and can be used to store and timestamp code and files during preregistration.[13] There is also a more comprehensive preregistration form available from the Open Science Foundation.[14] We describe a simple format for an experiment log which can be kept as a record of the experiment steps being executed. It can also be used to record any unavoidable changes that had to be made during final execution, e.g., if a code script had a bug that was not detected during testing and had to be fixed after the pre-registration.

Whilst the process for recruiting participants will be defined in the design and implementation, we briefly cover in Unit 7 issues that come up during experiment execution. In particular, we discuss how participants should be treated in an ethical and fair way. Whilst this covers all types of participant, we take a special look at crowd workers, such as those recruited via Amazon Mechanical Turk or Prolific, as there are unique issues which arise there due to the way the platforms operate and a lack of worker rights compared with conventional employment.

Finally, we take a brief look at post-final execution for reproducibility testing. This is a special case of execution where another team executes the experiment, ideally having access to all resources shown in the box on the right of Figure 3, with the aim of comparing similarity of results. After explaining the terminology used for reproducibility studies (Belz, 2022), we show how the degree of reproducibility between two or more studies can be measured. We do this by using the extended version of Quantified Reproducibility Assessment (QRA++) (Belz et al., 2022; Belz and Thomson, 2023, 2024) which supports degree of similarity assessments for four common types of results produced in NLP evaluations.

Unit 8: Extended Practical Session

The Extended Practical Session consists of two exercises covering: (i) Completion of preregistration forms and (ii) Analysis of results using Python.

In the first exercise, attendees select an experiment, either one of their own (it need not be published), or from the submitted reproduction papers by ReproHum project partners for the ReproNLP 2024 shared task (Belz and Thomson, 2024).[15] Attendees complete a dummy preregistration on aspredicted.org for a reproduction attempt of the selected paper. The exercise concludes with group discussion of issues encountered.

In the second exercise, attendees are guided through an analysis of results for one of the example experiments shown in earlier units. They will be able to follow along with the instructor, using Google Colab, as tools for data visualisation and statistical analysis are demonstrated.[16] For this

[13] https://github.com

[14] help.osf.io/article/145-preregistration

[15] https://aclanthology.org/events/humeval-2024 (those with "ReproHum" in the title).

[16] https://colab.research.google.com

exercise, we generate synthetic responses where LLMs act as evaluators rather than human participants.

3 Conclusion

In this paper, we have provided abstract-style summaries of each of the units of the INLG'24 Tutorial on Human Evaluation of NLP System Quality, also incorporating some of the core content from each of the tutorial's eight taught units.

The tutorial resources we provide via the tutorial's GitHub page (Footnote 1) of course provide more information and go into more detail on all of the aspects of human evaluation of NLP systems mentioned above, and this paper should not be considered complete or comprehensive in this respect.

We will set up a feedback mechanism for participants and offline users of our content, via the tutorial GitHub, to help us improve content and resources for future editions of the tutorial.

Acknowledgments

Mille's contribution was funded by the European Union under the Marie Skłodowska-Curie grant agreement No 101062572 (M-FleNS). Thomson's contribution was funded by the ADAPT SFI Centre for Digital Media Technology. Huidrom's PhD is funded by the Faculty of Computing and Engineering at Dublin City University. Our work on this tutorial has also benefited more generally from being carried out within the research environment of the ADAPT SFI Centre, funded by Science Foundation Ireland through the SFI Research Centres Programme and co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106.

References

Ron Artstein. 2017. Inter-annotator agreement. *Handbook of linguistic annotation*, pages 297–313.

Anya Belz. 2022. A metrological perspective on reproducibility in nlp. *Computational Linguistics*, 48(4):1125–1135.

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, Craig Thomson, and Rudali Huidrom. 2024. Qcet: An interactive taxonomy of quality criteria for comparable and repeatable evaluation of NLP systems. In *Proceedings of the 13th International Conference on Natural Language Generation*.

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

Anya Belz and Craig Thomson. 2023. The 2023 Repro NLP shared task on reproducibility of evaluations in NLP: Overview and results. In *Proceedings of the 3rd Workshop on Human Evaluation of NLP Systems*, pages 35–48, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Anya Belz and Craig Thomson. 2024. The 2024 Repro NLP shared task on reproducibility of evaluations in NLP: Overview and results. In *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval) @ LREC-COLING 2024*, pages 91–105, Torino, Italia. ELRA and ICCL.

Peter J Bickel and Kjell A Doksum. 2015. *Mathematical statistics: basic ideas and selected topics, volumes I-II package*. Chapman and Hall/CRC.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274, Online. Association for Computational Linguistics.

Christopher Chatfield. 1985. The initial examination of data. *Journal of the Royal Statistical Society: Series A (General)*, 148(3):214–231.

Liam Cripwell, Anya Belz, Claire Gardent, Albert Gatt, Claudia Borg, Marthese Borg, John Judge, Michela Lorandi, Anna Nikiforovskaya, William Soto-Martinez, et al. 2023. The 2023 webnlg shared task on low resource languages overview and evaluation results (webnlg 2023). In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Rotem Dror, Lotem Peled-Cohen, Segev Shlomov, and Roi Reichart. 2020. *Statistical significance testing for natural language processing*. Springer.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *Journal of Artificial Intelligence Research*, 77:103–166.

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.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Patrick Mineault and The Good Research Code Handbook Community. 2021. patrickmineault/codebook: 1.0.0.

Jordan Rodu and Michael Baiocchi. 2023. When black box algorithms are (not) appropriate. *Observational Studies*, 9(2):79–101.

Jordan Rodu and Karen Kafadar. 2022. The q–q boxplot. *Journal of Computational and Graphical Statistics*, 31(1):26–39.

Paul Rottger, Bertie Vidgen, Dirk Hovy, and Janet Pierrehumbert. 2022. Two contrasting data annotation paradigms for subjective NLP tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 175–190, Seattle, United States. Association for Computational Linguistics.

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.

Anastasia Shimorina, Claire Gardent, Shashi Narayan, and Laura Perez-Beltrachini. 2018. *WebNLG challenge: Human evaluation results*. Ph.D. thesis, Loria & Inria Grand Est.

Karen Spärck Jones, editor. 1981. *Information Retrieval Experiment*. Buttersworth.

Craig Thomson and Anya Belz. 2024. (mostly) automatic experiment execution for human evaluations of NLP systems. In *Proceedings of the 17th International Conference on Natural Language Generation*, page tbd, Tokyo, Japan.

Craig Thomson, Ehud Reiter, and Anya Belz. 2024. Common flaws in running human evaluation experiments in NLP. *Computational Linguistics*, pages 1–11.

Edward R Tufte and Peter R Graves-Morris. 1983. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.

L Wasserman. 2013. All of statistics: a concise course in statistical inference.

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.