Process Standardisation for Human Evaluation of NLP System Outputs

Craig Thomson[1], Javier González Corbelle[2], Anya Belz[1]
[1]ADAPT, Dublin City University
[2]CiTIUS, Universidade de Santiago de Compostela, Spain

Corresponding authors: craig.thomson@dcu.ie, anya.belz@dcu.ie

Abstract

Human evaluation of NLP systems has high knowledge and effort thresholds. Researchers are often expected to design and run evaluations without formal training, while also creating the required resources from scratch. Recent work has started to address the knowledge threshold, but reusable tools that reduce effort remain limited. In this paper, we take a first step toward automated human-evaluation experiment creation by (i) surveying the processes and data resources used in a representative sample of current human evaluations in NLP, and (ii) deriving a canonical process model from these survey results, which (iii) provides a basis for standardised experiment design and automated toolkit development. The survey shows that recent human-evaluation practices are highly aligned in process structure, making reusable automation feasible.

1 Introduction

A growing body of work paints a concerning picture of the state of human evaluation in NLP. Experiments have low reproducibility (Belz et al., 2023), statistical methods are applied haphazardly (Hämäläinen and Alnajjar, 2021), experiments are underpowered (Card et al., 2020; Howcroft and Rieser, 2021). Details of experimental design and implementation frequently go unreported, with resources as basic as the evaluation interface and the items shown to evaluators often not shared (Karpinska et al., 2021; Belz et al., 2023; Ruan et al., 2024; Schmidtova et al., 2025). When details are made available, many experiments are found to contain mistakes in their execution (Thomson et al., 2024).

Given such issues, it is hard to draw reliable conclusions from current human evaluations. Part of the reason for the parlous state of human evaluation in NLP is that researchers often have no prior experience or expertise that would equip them for designing and running human evaluation experiments. Moreover, they are expected to implement every aspect of an evaluation experiment from scratch. Recent research is beginning to address the high knowledge threshold with recommended design and techniques (van der Lee et al., 2021), standardisation of quality criteria (Howcroft et al., 2020; Belz et al., 2025b), tutorials (Belz et al., 2024), and papers identifying appropriate use of statistical and other methods (Card et al., 2020). However, given the absence of existing reusable tools for human evaluations, effort thresholds remain high.

In this paper, our aims are (i) to systematically survey the process structures of existing human evaluations in NLP; (ii) to determine whether the experiments' individual process structures can be merged into a single overarching structure that captures their commonalities while minimising redundancy; and (iii) to derive a standardised process structure that can in turn form the basis for extensive automation. In combination, this would lead to much improved human evaluation practices through lowering both knowledge and effort thresholds.

The remainder of this paper proceeds as follows. Section 2 presents the systematic survey. Section 3 derives a canonical process model and Section 4 describes how it can be used. Section 5 concludes.

2 Systematic Survey

We performed a systematic survey of the processes used for the evaluation of NLP system outputs in 100 papers sampled from ACL and EMNLP proceedings from 2018 to 2024.[1] We use the term **process** in the narrow sense of a discrete operation that transforms one or more data artefacts, e.g., evaluation items, participant lists, or responses, into one or more other data artefacts. Note that this definition excludes non-data-transforming processes such as participant recruitment.

[1]EMNLP in 2019 was combined with IJCNLP.

Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM), pages 704–717

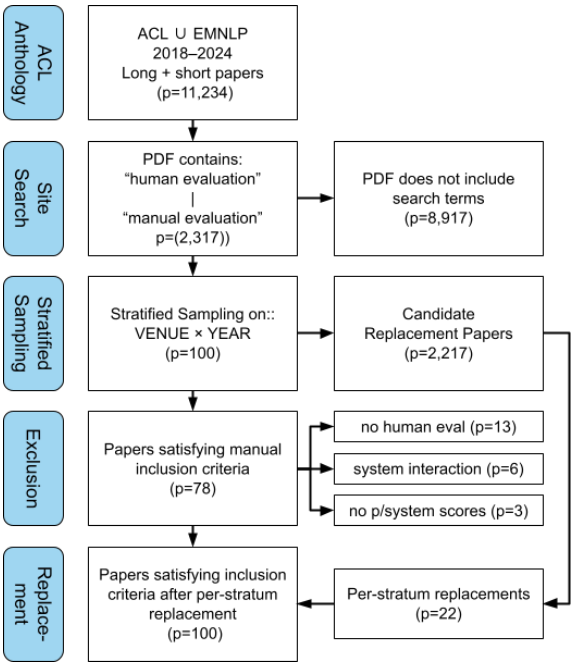

Figure 1: Sampling process graph.

2.1 Paper Sampling

As shown in Figure 1, we started by extracting all papers containing either the phrase "human evaluation" or "manual evaluation" from the ACL and EMNLP 2018–2024 proceedings via a Google site search on the ACL Anthology. This resulted in a set of 2,317 papers, from which we sampled 100 papers by stratified sampling on conference and year. The remaining 2,217 papers were retained as candidate replacements for excluded papers. Next we manually checked the 100 papers for the following three exclusion criteria:

1. **No human evaluation:** The paper includes the search phrase, but no human evaluation, e.g., a survey paper or a paper planning a human evaluation in future work.
2. **No per-system scores reported:** The human evaluation does not report aggregated system-level scores, e.g., because it is used for downstream correlation with metrics only.
3. **User-system interaction:** Users interact with the system, and therefore the evaluation items (NLP system outputs) are not available in advance, e.g., the user chats with a chatbot.

After excluding papers that met the exclusion criteria we were left with 78 papers: 13 papers were excluded for not containing a human evaluation, 3 papers did not report per-system scores, and 6 involved user-system interaction. We replaced each of the 22 excluded papers with the next random paper that met the inclusion criteria and was from the same conference × year stratum in the 2,217 retained papers. In all but one case, the first random paper from the same stratum was usable; the exception was one replacement paper that failed on the third criterion above, i.e., the evaluation it reported involved user-system interaction, but we were able to use the next candidate instead.

We were able to use 100 of the first 110 papers (or 91%) randomly sampled from all papers confirmed to include a human evaluation in ACL/EMNLP 2018–2024. This low exclusion rate, combined with our stratified sampling method and the sample size, means we can have confidence in the representativeness of our sample. In particular, our sample is likely to contain the most common types of human evaluation in the NLP literature.

The 100 papers in our final sample were split roughly evenly between ACL (48) and EMNLP (52), with more papers from more recent years (reflecting the rapid increase in paper numbers year on year in the NLP field over the period), as shown in Figure 2. We additionally sampled 24 papers for use in development and pilot annotations.

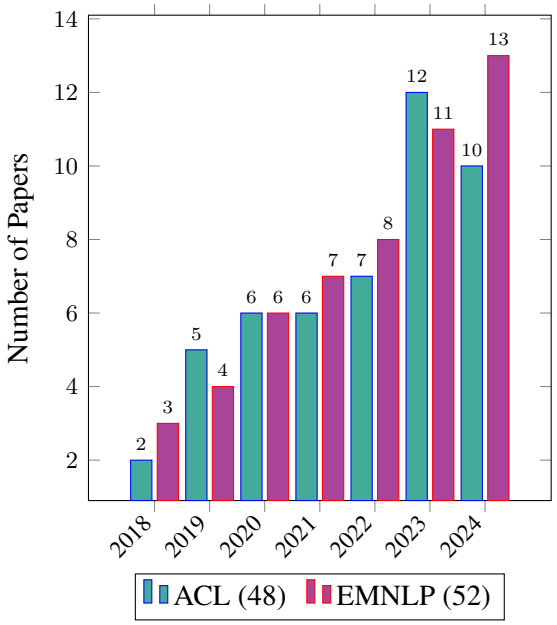

Figure 2: Number of papers selected by stratified sampling from ACL and EMNLP, between 2018 and 2024.

2.2 Annotation Protocol Development

We developed an annotation protocol where the component processes (e.g., data preprocessing, response collection) in experiments are drawn as

nodes in directed graphs. The protocol is focused on *what happens to the data*. We do not cover the important but separate issues of selecting quality criteria (Belz et al., 2025a), evaluation method and rating instruments (Popp et al., 2025), and participant recruitment (van der Lee et al., 2021).

Development had four phases: free annotation, consensus protocol creation, pilot annotation, and protocol revision. We describe each of these phases over the next four subsections.

2.2.1 Free annotation

During the initial phase, the first two authors, henceforth 'the annotators,' read the same ten ACL/EMNLP papers and independently annotated the experiment processes found in them in whichever way they felt was best. Both annotators have prior experience in human evaluation design and execution in NLP. One annotator chose text files and a custom format (for an example see Figure 3), whereas the other annotator used the Dia drawing tool[2] to create graphs (see Figure 4 for the same paper annotated by this method).

```
1. SYSTEM INPUTS
   || SYSTEM OUTPUTS
2. SAMPLE ITEMS [RANDOM]
3. ALLOCATE EVALUATION ITEMS PER PARTICIPANT
4. INTERFACE DESIGN
5. RESPONSE COLLECTION
6. RESPONSE AGGREGATION
   || STATISTICAL TESTING [KENDALL]
7. RESULTS
8. RESULTS PRESENTATION [TABLE]
```

Figure 3: Example of the text-based annotation method recording experiment processes that was chosen by one of the annotators.

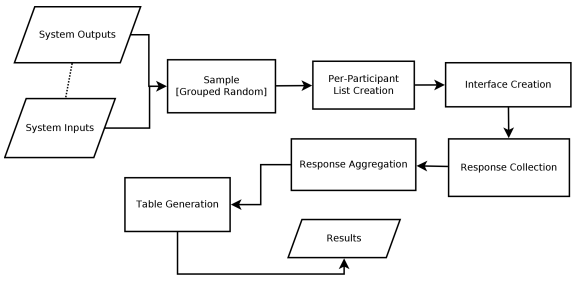

Figure 4: Example of the graph-based annotation method recording experiment processes that was chosen by one of the annotators. Note this annotator missed the use of a statistical test.

After discussing the free annotations, it was agreed that both methods were fundamentally capturing the same details, but that the drawn graphs would be more extensible and easier to analyse automatically.

2.2.2 Consensus annotation protocol

Based on these initial annotations, a standardised list of 10 named processes was created: `Item Sampling`, `Evaluation Item List Creation`, `Per-participant List Assignment`, `Interface Creation`, `Response Collection`, `Response Aggregation`, `Table Generation`, `Chart Generation`, `Agreement`, and `Statistical Test`.

It was also agreed that all graphs would start with an additional start node labelled `Candidate Evaluation Items`, and terminate with an end node labelled `Results`. Annotators were permitted to add new names if they encountered something not present in the standard list, and these would be standardised after each round (pilot and final) of annotations was complete.

In addition to recording the standardised node name for each process, the annotators also recorded *process implementation details* as a comma-separated list within square brackets after the node name, e.g., `Item Sampling [Random]`. This was done for all processes except for `Evaluation Item List Creation` and `Per-Participant List Assignment`. The reasons for not annotating these are (i) prior work has shown finding this information is seldom possible, and (ii) these processes inherently use the same simple combinatorics based on the total numbers of lists, items, and participants, and the relationships between them. Knowing the exact numbers would not tell us anything more about the processes.

The resulting annotation protocol was as follows. For each experiment reported in each paper:

1. Identify distinct human evaluation experiments in the paper.
2. For each experiment, create a new diagram in the Dia tool.
3. Draw the `Candidate Evaluation Items` start node.
4. From details reported in the paper only (including appendix) draw the specified processes, connecting them to form a graph, and also completing the list of process implementation details. If a process is found that is not named in the standard list, add it to the

[2] https://gitlab.gnome.org/GNOME/dia

list. Any process, except the start and termination nodes, may be used more than once in a diagram.

5. Draw the `Results` termination node. All processes that do not yet have output arrows should point to this node.

We had anticipated that some experiments would require conditional logic, and provided diamond-shaped nodes to represent it. However, we found no examples of loops or other conditional logic.

2.2.3 Pilot

We performed a pilot annotation to validate the protocol and determine agreement between the two annotators, with both annotating the same set of 14 papers, one from each year (2018–2024) from both ACL and EMNLP.

Both annotators identified the same 16 distinct experiments from the 14 papers. The process diagrams they drew were identical in eight cases. A further five differed only in that one annotator forked `Statistical Test` and `Annotator Agreement` after the `Response Aggregation` stage, whilst the other did so after the `Response Collection` stage.

Since these five cases differed only subtly, we consider these to be in agreement as well, and 13 of 16 graphs to be in complete agreement. In each of the final three cases, one annotator had missed a statistical test that was reported in the paper.

2.2.4 Protocol revision

No additional node labels were identified during the pilot. However, the use of `Response Collection` and `Response Aggregation` was clarified. For consistency, downstream processes would be connected to the former, unless they required the exact type of aggregation specified by the latter.

For the *process implementation details* recorded in brackets against most nodes, the annotators discussed their annotations and created a set of standardised labels, e.g., "Random", "Stratified", and "Unknown" for `Item Sampling`. As with the node names, annotators can create additional labels if needed. A palette of nodes with standardised additional information is shown in Figure 9 in the appendix as an illustrative reference; understanding the protocol does not depend on that figure.

2.3 Graph annotation of 100-paper sample

Graph Annotation was conducted on the main sample of 100 papers as per the adjusted protocol from the pilot (the 24 papers used during development were discarded). Each annotator created process graphs for 50 papers, split evenly between them by year and conference. Figure 5 shows the process graph created for Spangher et al. (2024) as a relatively simple example.

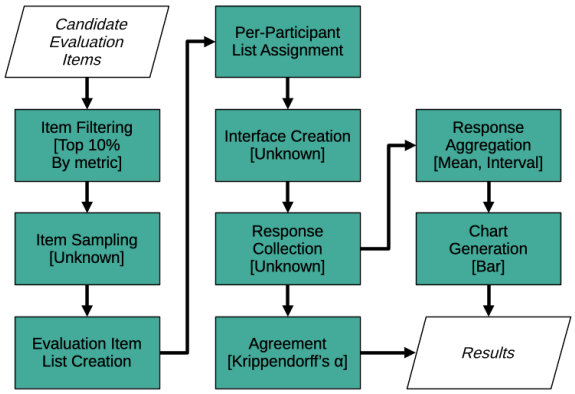

Figure 5: Process graph for Spangher et al. (2024) including a fork for where collected responses are used (a) to calculate inter-annotator agreement, and (b) to generate a chart. The type of interval calculated during response aggregation was not specified.

Figure 6 is an example of a more complex graph with multiple forks, created for Potluri et al. (2023): here, collected responses are passed on to (i) the `Agreement` process, and (ii) the `Table Generation` process, for two different methods of response aggregation (the per-criterion scores, and the percentage of summaries that are acceptable by all criteria). There is also a fork for sampling, because the authors "sample 150 examples at random and additionally sample 25 examples where the decontextualization process made edits to the gold extractive summary." Therefore there are two `Item Sampling` nodes, with the union of them used to create evaluation item lists.

2.4 Results

A total of 109 directed process graphs were created for the 100 surveyed papers. Of these, 42 (38.5%) are isomorphic to the most frequent graph (the central path in Figure 7). The mean Graph Edit Distance (Sanfeliu and Fu, 1983) between all graphs and this most common isomorph is 2.36, meaning that only around two edit operations are required to convert an observed graph to the most common isomorph.[3] However, this does not de-

[3]We used the Python nx implementation, (Abu-Aisheh et al., 2015) to compute graph edit distance. The mean GED remains low (3.84) when considering only graphs that do not conform to the most common isomorph.

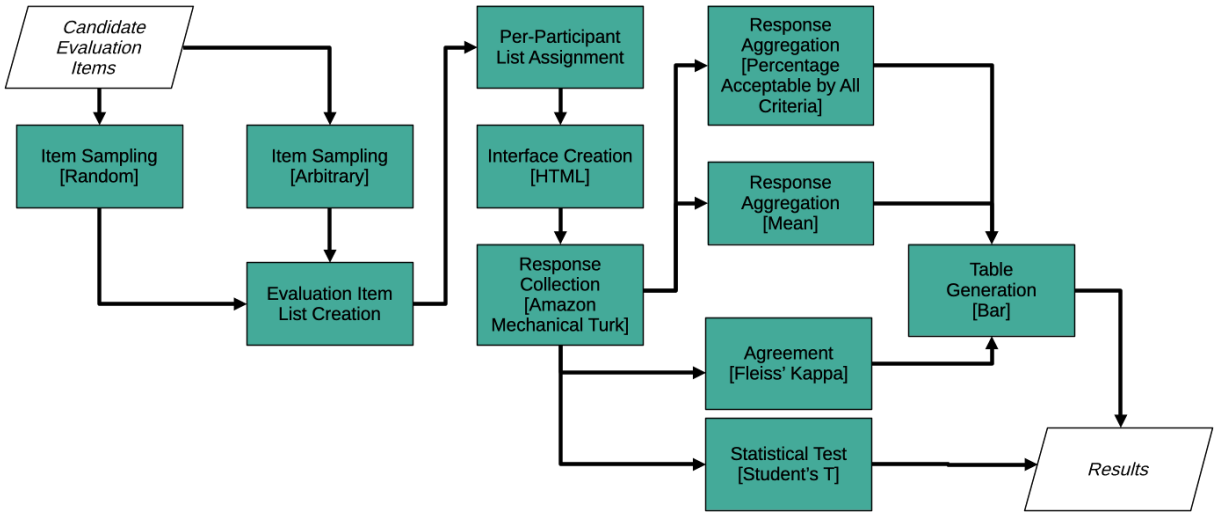

Figure 6: The experiment processes from Potluri et al. (2023) were annotated as a more complex graph, with multiple sampling nodes, as well as two methods of response aggregation.

scribe the ways in which the graphs differ. The most common differences were that (i) some papers performed a statistical analysis, others didn't; (ii) some calculated inter-annotator agreement, others didn't; and (iii) some presented results as tables, some as charts, and some used neither (reporting them inline in text instead). These common differences accounted for a further 31 (28.4%), with 36 (33.0%) differing in other ways.[4]

Figure 7 shows the union graph for the 109 process graphs resulting from the annotation. The thickness of the vertices in the union graph indicates how many of the 109 graphs had a given edge (connecting the same two vertices), with line width proportional to graph count; dashed lines indicate counts of 1–2. The most common isomorph is readily detectable as the central column of light green nodes connected by the thickest edges. Although exactly 42 graphs are isomorphic to this form, many additional graphs contain this same core path with extra branches, which is why central edge counts can be much higher (e.g., 87 graphs include `Table Generation` → `Results`).

2.4.1 Process implementation details

Annotation included noting down implementational details for each process where provided in a paper (Section 2.2.2). We summarise our findings below for each process, with complete results and discussion provided in Section A.1.

Item Sampling: Of 109 experiments, 59 used random sampling, while no information was provided for 43. The tail included quota sampling, first N, and one case where participants informed the process by selecting categories of news articles they would be interested in (Cai et al., 2023).

Item Filtering: Only found in 4 of 109 experiments, e.g., Wu et al. (2021) removed evaluation items where the system input resulted in a set of system outputs that varied too greatly in length.

Item Processing: This was only used in one case, by Fan et al. (2019) who truncate generated stories at the sentence boundary nearest 200 words.

Evaluation Item List Creation: All experiments perform this by definition, based on the selection criteria. If no intentional list creation takes place, the result is a list containing all evaluation items.

Per-Participant List Assignment: As above, performed by definition. If no intentional assignment takes place, the result is a set (per-participant) of the single list of all items.

Interface Creation: Only 15 experiments indicated the type of interface used, of which 13 were HTML. The remaining 94 provided no details.

Response Collection: For 70 of 109 experiments, the method of response collection, i.e., the platform or method by which participants entered responses, was not given. The most common given method was Amazon Mechanical Turk (26), followed by unspecified crowd platforms (4).

Response Exclusion: Only Kreiss et al. (2022) reported any response exclusion. They exclude e.g., all responses from participants who failed attention checks or self-reported not feeling they correctly completed the task.

Statistical Testing: Only 21 of the 109 experiments performed statistical tests, with one exper-

[4]Percentages do not sum to 100.0 due to rounding.

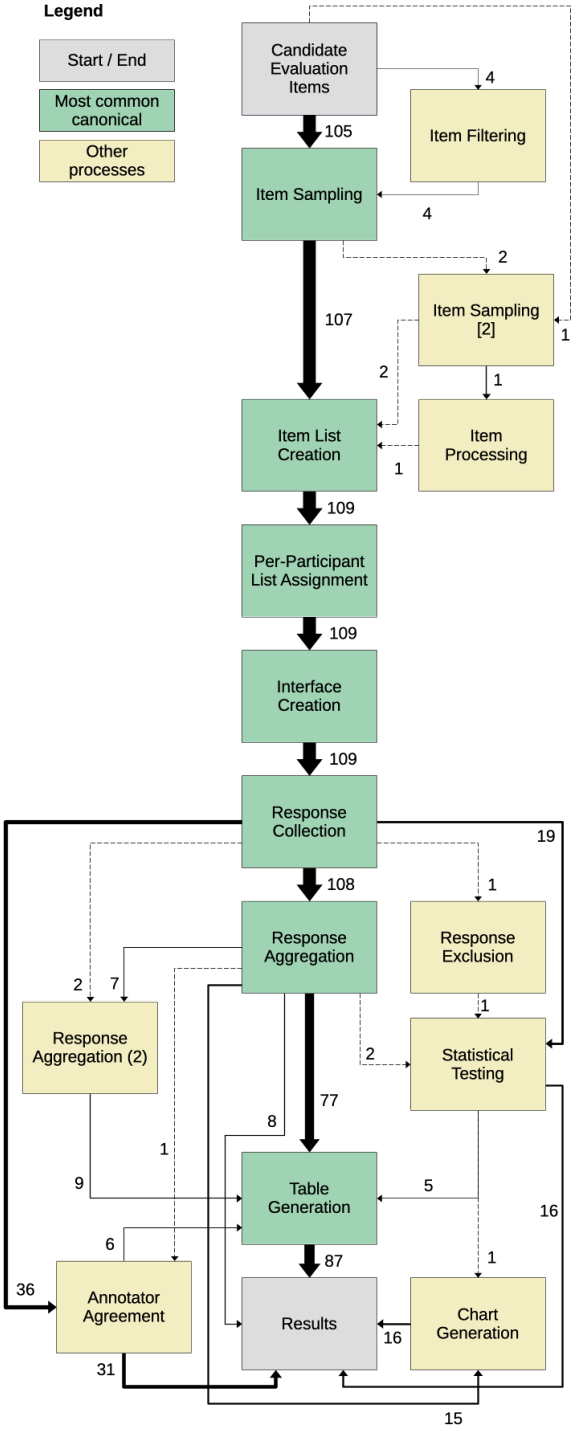

Figure 7: Union of process graphs from the main annotation exercise (100 papers). The most common isomorph is the central path from `Candidate Evaluation Items` to `Results`. 42 graphs (38.5%) are exactly of this form.

iment performing two tests. No specific test was used with any great frequency. Pearson's Correlation was used by 6 experiments.

Annotator Agreement: Of the 37 experiments where agreement was reported, the most common methods were Fleiss' κ (11), Cohen's κ (6), and Krippendorff's α (10).

Response Aggregation: Most common were mean score (43 experiments) and percentage (27), the latter usually for binary responses. 18 experiments aggregated in a win-loss format, some with ties, as well as 10 that simply report raw counts from ordinal or categorical rating instruments.

Chart Generation: 16 experiments presented results in charts; 14 as bar charts, 2 as scatter plots.

Table Generation: Tables were by far the most common way of presenting results (87 experiments). All tables reported per-system scores, i.e., there were no confusion matrices etc. 6 only presented results inline in the paper (no tables/charts).

Changes in processes used over time. We looked at the inclusion of statistical testing and annotator agreement processes by publication year (2018–2024). Across the 100-paper sample, statistical testing appeared in about one third of papers from 2018 and 2022–2024, but was absent entirely in 2019–2021. Annotator agreement was more common in recent years, rising from around 25% in 2020–2022 to 39.1% in 2023 and 60.9% in 2024. Table-based reporting was the most common approach in all years, used in at least two thirds of papers. Given stratified sampling and modest per-year counts, we treat these as descriptive patterns rather than inferential trend claims.

3 Canonical Process Model

While the surveyed experiments vary in their implementational details, their process graphs are strikingly similar. Most have a small number of processes applied to data resources such as evaluation items and participant lists. These processes typically form a simple pipeline: candidate items are sampled, arranged into evaluation lists, presented to participants through an interface, and the resulting responses are aggregated and used to generate a table. Variations are usually minor insertions of additional processes (for example, statistical testing or agreement calculation) rather than a fundamentally different graph.

Motivated by this consistency, we derive a **canonical process model** that captures the common structure of these experiments, starting from the most common isomorph. The model represents evaluations in terms of persistent data resources and the processes that transform them. This abstraction allows human evaluation experiments to be described as sequences of discrete operations

over well-defined data resources, providing a foundation for both reproducibility and automation.

Figure 8 presents the canonical process model. Each node represents either a persistent data resource (usually the output from one process and the input to the next), or a process that transforms one or more resources into another. This descriptive model aligns with that of Belz et al. (2024), but further breaks down some of their *Component Processes* into multiple discrete processes, on the basis of our empirical survey.

Given a small number of simple standardisations, this model can be used to describe all observed experiments. Definitions for each data resource (ovals in Figure 8), and the processes that create them (oblongs in Figure 8), are given below. For brevity, `Evaluation Items` are referred to simply as `Items`.

3.1 Canonical data resources

`Candidate Items`: Each item comprises an input from the dataset, and any corresponding outputs from systems being evaluated.

`Processed Items`: The items resulting from the application of `Item Processing`, i.e., one or more functions that change individual items. For example, system outputs may be lower-cased, or truncated at N words.

`Filtered Items`: The items resulting from `Item Filtering`, e.g., removing items for which one of the model responses was NULL.

`Sampled Items`: The items resulting from `Item Processing` where `Filtered Items` are selected with a predefined sampling strategy.

`Item Lists`: The list of items resulting from the application of `Item List Creation` where items are assigned to one or more lists based on combinatorics from the experiment design. For example, there may be 192 items in the sample, split into 6 lists of size 32.

`Per-Participant Lists`: The lists of items resulting from the application of `Per-participant List Creation` where items are assigned to participant pseudonyms. For example the design may specify 3 participants per item, so continuing the above example, a copy of each list is created and assigned to participants `P001` through `P018`.

`Populated Interfaces`: The items resulting from the application of `Interface Population` to `Per-participant List` items where an interface template is populated for each real participant.

`Collected Responses`: The items resulting from applying the `Response Collection` process to `Populated Interface` items, where responses are extracted from each interface e.g., spreadsheet or web application form.

`Statistical Significance Results`: The items resulting from the application of the `Statistical Significance Testing` process, i.e., the results from one or more predefined statistical significance tests, along with a record of any corrections for multiple hypothesis testing.

`Agreement Results`: Inter/intra-annotator agreement calculated from `Collected response` items by the `Annotator Agreement` process.

`Aggregated Responses`: Responses aggregated by the `Response Aggregation` process in order to obtain per-system scores. For example, through `COUNT` or `MEAN` operations.

`Generated Tables`: Tables generated from `Responses` and/or `Results` items above by a `Table Generation` process.

`Generated Charts`: Charts generated from `Responses` and/or `Results` items by a `Table Generation` process.

3.2 Canonical processes

`Item processing`: Transforms `Candidate Items` to `Processed Items`, e.g., normalising whitespace.

`Item filtering`: Transforms `Processed Items` to `Filtered Items` by applying given exclusion rules, e.g., removing items with inputs longer than a limit.

`Item sampling`: Transforms `Filtered Items` to `Sampled Items`, e.g., by stratified sampling on given properties.

`Item list creation`: Transforms `Sampled Items` to `Item Lists`, containing just the items to be used in the evaluation, keeping track of input, system, and other IDs.

`Per-participant list creation`: Transforms `Item Lists` to `Per-participant Lists`, where each contains just the items for one participant to evaluate. This is a standardisation that we make, requiring such lists to be available in advance, e.g., not allowing items to be served to evaluators on the fly.

`Interface creation`: Transforms `Per-participant Lists` and interface templates to `Populated Interfaces`, e.g., a Google form.

`Response collection`: Transforms `Populated Interfaces` to `Collected Responses`, typically by giving evaluators

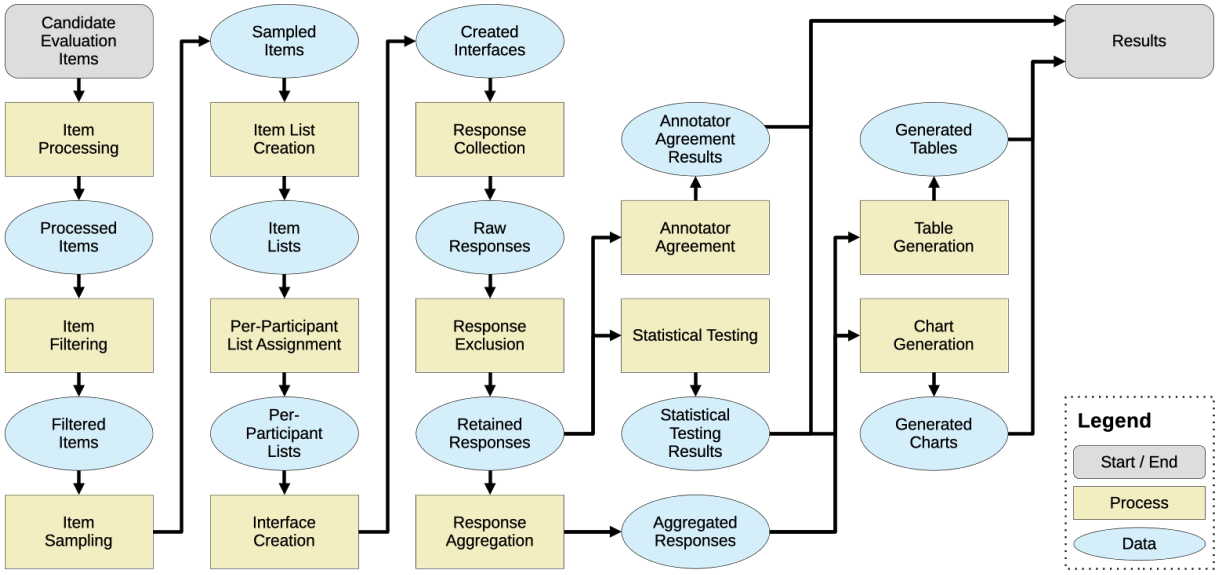

Figure 8: Canonical process model showing data resources and the processes that operate on them.

access to their individual populated interface to enter responses per item, which are then extracted into a structured response record.

`Response aggregation:` Transforms `Collected Responses` to `Aggregated Responses`, e.g., computing system-level means.

`Inter-annotator agreement estimation:` Transforms `Collected Responses` to `Agreement Results`, e.g., Fleiss's κ over paired annotations.

`Statistical significance testing:` Transforms `Collected Responses` to `Statistical Significance Results` by applying statistical tests such as an ANOVA or Kruskal-Wallis test.

`Table generation:` Transforms (one or more of) `Aggregated Responses`, `Agreement Results`, and/or `Statistical Significance Results` to `Generated Tables` presenting the corresponding values, e.g., a table of system-level mean scores with significance groups denoted by labels.

`Chart generation:` Transforms (one or more of) `Aggregated Responses`, `Agreement Results`, and/or `Statistical Significance Results` to `Generated Charts` graphically presenting the corresponding values, e.g., a bar chart showing mean scores for different types of systems.

4 Using the Canonical Process Model

4.1 Individual experiments

The immediate way in which the canonical process model (CPM) introduced in this paper can be used is in guiding the design of human evaluation experiments. The approach we propose for this consists of three steps:

1. Write a specification for each process in Figure 8, including definitions for the inputs and outputs, e.g., names and types for columns in a dataframe, as well as details of the process. Each process should be implemented as a function that takes as input one or more of the input resources and produces a new output resource without modifying prior files. If a process is to be omitted, e.g., no `Item Filtering` is to be performed, make this explicit by indicating "none performed."
2. Implement each of the processes such that they are clearly separate in the code. Use globally unique filenames for any data so that copies are kept of all files, do not update any files once they have been created.
3. Pre-register the experiment by uploading details on a preregistration website like AsPredicted.org, and by uploading code (privately), e.g., to GitHub.com.

Additional implementation notes:

- The associated `Item List`, `Items`, and `Participant` should be included in each response record. This may seem obvious, but published papers do exist where, e.g., the system output itself was (incorrectly) used to uniquely identify responses.
- `Agreement Results` and `Statistical Testing Results` form the input to `Table`

`Generation` or `Chart Generation` processes, but it is also prudent to retain the retain the raw output from each test, including e.g., residuals.

- Exact numeric values may not be determinable from a `Generated Chart`; a table in the appendix or supplementary material that contains these values enables reproducibility (Onderková et al., 2025).

Having and keeping separate input/output files documents data provenance which in turn supports recreatability and reproducibility. W3C defines provenance as "information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness."[5]

The above design steps complement the recording of the properties of the final experiment, e.g., using a Human Evaluation Data Sheet (Shimorina and Belz, 2022; Belz and Thomson, 2025).

4.2 Reusable tools

Once we have standardised processes and data resources, creating standard automated tools to make the creation of human evaluations easier and faster become much more straightforward.

To our knowledge, we do not currently have a general-purpose toolkit for conducting human evaluations of NLP systems, nor does any toolkit appear to be widely adopted even within specific subdomains. Kasner et al. (2024) introduce and demonstrate a framework for creating and analysing error annotations, and Thomson and Belz (2024) found only 1 of 20 experiments that provide data and code do so in a way that is close to end-to-end.[6] However, no approach is readily adaptable to the diversity of tasks, rating instruments, and evaluation criteria observed across NLP.

In this paper, our goal has been to standardise the processes already in use in human evaluations of NLP systems (as evidenced by our survey), and thereby provide the basis for researchers to create their own experiments in a way that supports rigorous recording of experimental details and procedures, and—because standardised—directly ensures comparability across different experiments.

In future work, we plan to create a toolkit to support maximally automated human evaluation creation, situated within a framework that will streamline this process and integrate other work on standardisation, such as that on evaluation measures (Belz et al., 2020), experiment documentation (Shimorina and Belz, 2022), quality criteria (Belz et al., 2025a), and UX design (Calò et al., 2025).

5 Conclusion

We have presented a systematic survey annotating the process structures used in the human evaluations found in 100 papers from the main conference proceedings of the two leading NLP conferences during 2018–2024. On the basis of the graphs produced in the annotation, we found that the majority of human evaluation experiments have a strikingly similar structure and exploited this fact to derive a canonical process model which can serve as a guide in creating new experiments that are comparable, recreatable and reproducible by design. Moreover, we have argued that the canonical process model provides the basis for creating modular, reusable code for a toolkit that maximally automates human evaluation experiments for NLP.

Our survey also yielded insights into current practice within each process, and echoed issues reported elsewhere: around four in five experiments do not report statistical significance tests, despite such testing being used in NLP evaluation for at least 30 years (Sparck Jones and Galliers, 1995) and repeatedly argued to be necessary (Yeh, 2000; van der Lee et al., 2019; Gehrmann et al., 2023). Inter-annotator agreement was also under-reported, with only one third of experiments including it.

Issues in human evaluation can at times seem unwieldy and intractable. However, better practices and resources are emerging. We provide a standard methodology for designing and implementing reusable code for human evaluation experiments. The methodology, derived from observed norms, is both simple and immediately actionable, mitigating many issues that roadblock the creation of reliable and reproducible human evaluations.

Limitations

Approximately 5% of ACL and EMNLP papers published between 2018 and 2024 containing the phrase "human evaluation" or "manual evaluation" were annotated in this study. Whilst the experiments examined were highly similar in structure,

[5] https://www.w3.org/TR/prov-dm

[6] Liu et al. (2021) provide a repository that includes two python notebooks, one for creating the human evaluation, and one to analyse the responses and generate charts.

differences may exist in unannotated studies.

We exclude studies in which participants directly interact with systems. Although such studies likely share certain components with the experiments surveyed here, particularly in terms of statistical analysis, the processes governing human interaction are typically more complex.

Our focus is on evaluations of NLP system outputs. Other forms of empirical study, such as evaluations of efficiency or impact, remain comparatively rare in the literature but do exist, and would likely involve additional and/or different processes.

Ethics

The annotation and analysis of existing peer-reviewed publications involves minimal ethical risk. Care was taken to not single out in a negative way any particular author of prior work. The issues around experimental design and reproducibility are ours as a field to address. Generative AI tools were used as an aid for word choice, phrasing, and code suggestion.

Acknowledgments

Thomson's work was funded by the ADAPT SFI Centre for Digital Media Technology. Our work has benefitted more generally from being carried out in the wider context of the ADAPT SFI Centre for Digital Media Technology which is 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. We would like to thank the reviewers and area chair for their constructive feedback.

References

Zeina Abu-Aisheh, Romain Raveaux, Jean-Yves Ramel, and Patrick Martineau. 2015. An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems. In *4th International Conference on Pattern Recognition Applications and Methods 2015*, Lisbon, Portugal.

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, and Craig Thomson. 2025a. The qcet taxonomy of standard quality criterion names and definitions for the evaluation of nlp systems. *Preprint*, arXiv:2509.22064. 39 pages, 7 figures.

Anya Belz, Simon Mille, and Craig Thomson. 2025b. Standard quality criteria derived from current NLP evaluations for guiding evaluation design and grounding comparability and AI compliance assessments. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26685–26715, Vienna, Austria. Association for Computational Linguistics.

Anya Belz, João Sedoc, Craig Thomson, Simon Mille, and Rudali Huidrom. 2024. The INLG 2024 tutorial on human evaluation of NLP system quality: Background, overall aims, and summaries of taught units. In *Proceedings of the 17th International Natural Language Generation Conference: Tutorial Abstract*, pages 1–12, Tokyo, Japan. Association for Computational Linguistics.

Anya Belz and Craig Thomson. 2025. HEDS 3.0: The human evaluation data sheet version 3.0. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 60–81, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Anya Belz, Craig Thomson, Ehud Reiter, and Simon Mille. 2023. 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.

Pengshan Cai, Kaiqiang Song, Sangwoo Cho, Hongwei Wang, Xiaoyang Wang, Hong Yu, Fei Liu, and Dong Yu. 2023. Generating user-engaging news headlines. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3265–3280, Toronto, Canada. Association for Computational Linguistics.

Eduardo Calò, Lydia Penkert, and Saad Mahamood. 2025. Lessons from a user experience evaluation of NLP interfaces. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2915–2929, Albuquerque, New Mexico. Association for Computational Linguistics.

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.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–

2660, Florence, Italy. Association for Computational Linguistics.

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.

Mika Hämäläinen and Khalid Alnajjar. 2021. The great misalignment problem in human evaluation of NLP methods. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 69–74, Online. 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.

David M. Howcroft and Verena Rieser. 2021. What happens if you treat ordinal ratings as interval data? human evaluations in NLP are even more underpowered than you think. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8932–8939, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using Mechanical Turk to evaluate open-ended text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1265–1285, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zdeněk Kasner, Ondrej Platek, Patricia Schmidtova, Simone Balloccu, and Ondrej Dusek. 2024. factgenie: A framework for span-based evaluation of generated texts. In *Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations*, pages 13–15, Tokyo, Japan. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Elisa Kreiss, Fei Fang, Noah Goodman, and Christopher Potts. 2022. Concadia: Towards image-based text generation with a purpose. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4667–4684, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. 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 6691–6706, Online. Association for Computational Linguistics.

Kristýna Onderková, Mateusz Lango, Patrícia Schmidtová, and Ondrej Dusek. 2025. ReproHum #0669-08: Reproducing sentiment transfer evaluation. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 601–608, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Birgit Popp, Sarah Keck, Androniki Mertsiotaki, Emily Kratsch, and Alexander Daum. 2025. Which method(s) to pick when evaluating large language models with humans? - a comparison of 6 methods. Fraunhofer Publica Preprint. Preprint.

Abhilash Potluri, Fangyuan Xu, and Eunsol Choi. 2023. Concise answers to complex questions: Summarization of long-form answers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9709–9728, Toronto, Canada. Association for Computational Linguistics.

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.

Fahime Same, Guanyi Chen, and Kees Van Deemter. 2022. Non-neural models matter: a re-evaluation of neural referring expression generation systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5554–5567, Dublin, Ireland. Association for Computational Linguistics.

Alberto Sanfeliu and King-Sun Fu. 1983. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(3):353–362.

Patricia Schmidtova, Eduardo Calò, Simone Balloccu, Dimitra Gkatzia, Rudali Huidrom, Mateusz Lango, Fahime Same, Vilém Zouhar, Saad Mahamood, and Ondrej Dusek. 2025. Do my eyes deceive me? a survey of human evaluations of hallucinations in NLG. In *Proceedings of the 18th International Natural Language Generation Conference*, pages 60–79, Hanoi, Vietnam. 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, Dublin, Ireland. Association for Computational Linguistics.

Alexander Spangher, Nanyun Peng, Sebastian Gehrmann, and Mark Dredze. 2024. Do LLMs plan like human writers? comparing journalist coverage of press releases with LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21814–21828, Miami, Florida, USA. Association for Computational Linguistics.

Karen Sparck Jones and J. R. Galliers. 1995. *Evaluating Natural Language Processing Systems: An Analysis and Review*, volume 1083 of *Lecture Notes in Computer Science; Lecture Notes in Artificial Intelligence*. Springer, Berlin. Includes bibliographical references (p. 219–225) and index.

Craig Thomson and Anya Belz. 2024. (mostly) automatic experiment execution for human evaluations of NLP systems. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 272–279, Tokyo, Japan. Association for Computational Linguistics.

Craig Thomson, Ehud Reiter, and Anya Belz. 2024. Common flaws in running human evaluation experiments in NLP. *Computational Linguistics*, 50(2):795–805.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

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.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Ziqiang Cao, Sujian Li, Hua Wu, and Haifeng Wang. 2021. BASS: Boosting abstractive summarization with unified semantic graph. 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 6052–6067, Online. Association for Computational Linguistics.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

A Appendix

A.1

In this section, we show all properties that were recorded for all types of process node. Note that these are counts of occurrences in graphs (a small number of papers had two graphs) and that one process node can have more than one property, e.g., a chart can have both bars and 95% Confidence intervals.

Item Filtering: This type of node was only seen 4 times. The method and reasons for filtering varied; one paper selected items based on a metric, another excluded input items with certain properties, a third excluded duplicate outputs, and the final paper only selected sets of system outputs that belonged to the same input, but were significantly different from each other (the method for determining this threshold was not given).

Item Sampling: Table 1 shows the property counts for Item Sampling. Random sampling was the most common method, although a large number of papers did not report a method at all.

Property	Count (graphs)
Random	59
Unknown	43
Quota	6
Arbitrary	1
Participant Cat. Selection	1
First N	1
> 150 words	1

Table 1: Property counts for Item Sampling nodes.

Item Processing: This node was only seen once, with system outputs being truncated to the nearest sentence boundary to 200 words (Fan et al., 2019).

Interface Creation: Table 2 shows the counts of observed interface types, with the vast majority (94 of 109) being unknown. Of the known types, HTML was the most common (13), usually as an Amazon Mechanical Turk template.

Property	Count (graphs)
Unknown	94
HTML	13
Google Forms	1
Appen	1

Table 2: Property counts for Interface Creation nodes.

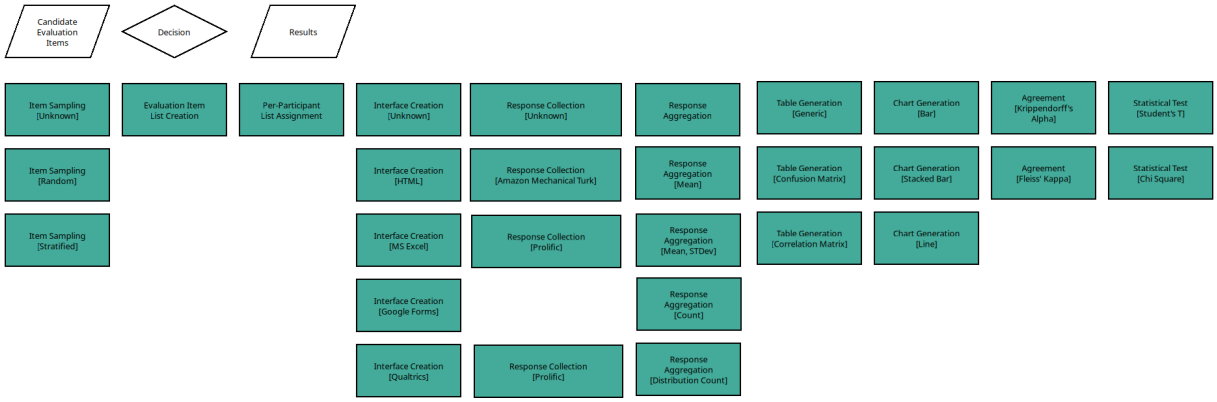

Figure 9: This palette of nodes / additional information was used by the annotators in the Dia application, i.e., they would copy and paste from this, and only draw nodes from scratch or add new types of additional information in brackets if required.

Response Collection: Table 3 shows the platform used to collect responses was unknown in most cases, with Amazon Mechanical Turk being the most commonly given platform.

Property	Count (graphs)
Unknown	70
Amazon Mechanical Turk	26
Crowdsourcing	4
Upwork	2
Google Forms	1
SurgeHQ	1
Appen	1
Prolific	1

Table 3: Property counts for Response Collection nodes.

Response Aggregation: This node type had the most diverse set of properties and was included in many papers (it is highly unusual to present results without aggregating them). Table 4 shows the property counts, with mean being the most common, with percentages, as well as results presented in a win-loss-tie format (for relative evaluations) also being common. One of the more unusual methods was reporting the percentage of wins plus ties (together rather than separately).

Table Generation: All tables were generic (each row reporting a per-system value). We saw one table where superscript letters had been used to denote significance groups (Same et al., 2022).

Chart Generation: Whilst charts were less common than table for presenting results, their type was more varied, with bar charts being the most common. The types seen are shown in Table 5. Note that the one observed 95% confidence interval was attached to a bar chart.

Property	Count (graphs)
Mean	43
Percentage	27
Win-Loss Percentage	11
Count	10
Standard Deviation	6
Win-Loss-Tie Percentage	7
Mean Rank	3
Pass-Fail Percentage	3
Majority Vote	2
Rank Count	2
Percentage (sum wins plus ties)	1
Pct. Acceptable by All Criteria	1
All Agree	1
Best-Worst Scaling	1
Unspecified Interval	1
Normalization	1
Sum	1
Weighted product	1
Collapse scores	1

Table 4: Property counts for Response Aggregation nodes.

Property	Count (graphs)
Stacked Bar	7
Bar	7
Scatter Plot	2
95% CI	1

Table 5: Property counts for Chart Generation nodes.

Statistical Testing: Statistical methods were rarely used. When they were, the type varied, which makes sense given that different tests will be appropriate for different types of response data. Table 6 shows the types seen. "Unknown" indicated that the paper mentioned significance testing, but did not give a method.

Property	Count (graphs)
Pearson Correlation	6
Unknown	4
Student's T	2
Z-Test	1
Wilcoxon Rank Sum	1
Paired T Test	1
Two-sample T-test	1
Wilcoxon signed rank (+Bonferroni)	1
Baysian mixed effects regression	1
(Koehn, 2004)	1
95% CI	1
posthoc Tukey HSD	1
One-way ANOVA	1
Phi correlation coefficient	1
Kendall's Tau	1

Table 6: Property counts for Statistical Test nodes.

Annotator Agreement: Also rarely used, the methods are listed in Table 7

Property	Count (graphs)
Fleiss' Kappa	11
Krippendorff's Alpha	10
Cohen's Kappa	6
Pearson Correlation	3
Unknown	3
Simple Majority	1
Kappa (unspecified)	1
Quadratic Cohen's Kappa	1
Randolph's Kappa	1

Table 7: Property counts for Agreement nodes.

B Counting and normalisation.

The property counts in this appendix are *counts of occurrences in process-graph nodes*, not unique-paper counts. Because some papers have multiple process graphs, and because a single process node can contain multiple bracketed properties, totals in a table can exceed the number of papers/graphs where that node type appears.

To improve consistency across annotations, we normalize near-equivalent labels before counting. In particular:

- Pearson / Pearson's / Pearson's Correlation → Pearson Correlation
- Counts / Distribution Count → Count
- STDev / STdev / Stdev → Standard Deviation
- Quota (random) / Quota Random → Quota
- Win / percentage-preference variants → Win-Loss Percentage
- Win-Loss-Tie variants (e.g., `Win-Loss-Tie`, `Win-Loss-Tie Percentage`) → Win-Loss-Tie

Merging used for Figure 7. To improve readability, Figure 7 uses merged process labels for low-frequency duplicate nodes. In particular, enumerated duplicates such as `Statistical Test (2)` and `Table Generation (2)`, which represent additional tables or statistical tests within the same experiment, are merged into their base process types (`Statistical Test`, `Table Generation`). Likewise, we merge the response-exclusion sequence `Response Exclusion` → `Response Exclusion (2)` → `Statistical Test` into a single transition. In total, this merges **6** low-frequency relation occurrences (each observed once): `Response Collection` → `Statistical Test (2)`, `Statistical Test (2)` → `Results`, `Response Aggregation` → `Table Generation (2)`, `Table Generation (2)` → `Results`, `Response Exclusion` → `Response Exclusion (2)`, and `Response Exclusion (2)` → `Statistical Test`. We only apply merges where they preserve the same high-level process flow; relations that introduce a different structural path are not merged away. All counts reported for fig. 7 follow this merged representation.