Aura : A Self-Hostable Platform for Human Evaluation of NLG for Under-resourced Languages and Beyond

Rudali Huidrom[1], Khangembam Alex D Nelson[9], Nicholas Khumukcham[2], Hoomexsun Pangsatabam[3], Jesu Nazarene Roy L[4], Kanal Trivedi[9], Vikas Kumar [2], Pinaki Das[5,6], Kshetrimayum Boynao Singh[7,8], Tarun Bisht[5], Justice Konjengbam[2], Anand Konjengbam[2], Vikram Goyal[1]

[1]IIIT Delhi [2]IIT Dharwad [3]NIT Manipur [4]EFL University, Hyderabad [5]IIT Bombay [6]Monash University [7]NIT Silchar [8]IIT Patna [9]Independent Researcher

Correspondence: *huidromrudali@gmail.com*

Abstract

Human evaluation is the gold standard for Natural Language Generation (NLG), but the tooling is fragmented: most platforms serve a single task or a single protocol, and none treat the growing practice of *assisting annotators with an LLM* as a measurable object of study. These gaps fall hardest on *indigenous and under-resourced languages*, where paid crowds are scarce, references are unavailable, and data governance matters. We present **Aura**, an open, self-hostable platform that unifies five evaluation protocols (Likert/DA, pairwise, ranking, error-span, post-editing) across eight task families under one segment schema and one result model (our user study exercises ranking for MT; the others are implemented and demonstrated), records writing-system and speaker provenance as first-class fields, and runs entirely offline. Aura's contribution is integrative: it combines multi-protocol evaluation, speaker and script provenance, and reproducible results packaging with an advisory LLM assistant whose availability can be randomised and logged to measure its effect. We conducted a human evaluation experiment on three English→Indian language ranking campaigns with nine annotators who completed the task without AI assistance. We subsequently included three additional annotators with the AI assistant enabled to evaluate Aura's AI-assisted annotation functionality. Across the 12 participants, the platform received a mean System Usability Scale (SUS) score of **82.29**/100 (SD = **16.53**). Aura is released as open-source software.[1]

1 Introduction

Human judgement still remains decisive in verifying the quality of Natural Language Generation (NLG) , yet the protocols to analyse it, Direct Assessment (Graham et al., 2013), Multidimensional Quality Metrics (Freitag et al., 2021), error-span annotation (Kocmi et al., 2024), and pairwise preference (Bradley and Terry, 1952), each impose distinct interfaces, agreement statistics, and failure modes (Karpinska et al., 2021). Existing platforms mostly support one or a few protocols, so that a study is typically assembled from a spreadsheet, a crowd marketplace, or a purpose-written script, and reassembled for different protocols. The resulting workflows demand a lot to construct, reproduce, and do cross-study comparisons. Aura addresses this fragmentation by mapping five protocols and eight task families onto a single segment schema and a single result model, so that a change of protocol constitutes a change of configuration rather than of pipeline.

In the context of India, although machine translation has been extended to under-resourced languages, including all 22 scheduled Indian languages (Gala et al., 2023), rigorous human evaluation for these languages remains difficult to organize. The difficulty is sharpest for indigenous and under-resourced languages such as Manipuri, Nagamese, where there is often no paid crowd to draw on, annotation work proceeds through participatory, community-governed arrangements that sit uneasily with hosted SaaS products.

For a high-resource language, the components ignored by a platform can be procured elsewhere. For under-resourced languages, evaluation proceeds mostly under participatory, community-governed arrangements to which hosted platforms are poorly suited. The writing system, including the choice of script in digraphic cases, is itself under evaluation. Such studies require several campaigns, one per language pair, system, or difficulty level, whose outputs must then be compiled across files sharing neither annotator identifiers nor a schema. In Aura, a stable annotator UUID links an annotator's work across campaigns; compatible campaigns can be pooled into a *group*, and each campaign or group

[1]Live at https://www.aura4lrl.studio; demo video at https://youtu.be/NDtqi_Rch2I; GitHub: https://github.com/RHuidrom96/aura

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 9–16
November 6-10, 2026 ©2026 Association for Computational Linguistics

yields a single reproducible results pack.

LLMs increasingly enter the annotation loop, both as substitutes for annotators (Gilardi et al., 2023) and, more cautiously, as assistants to them (Wang et al., 2024). Using a model to gloss an idiom or explain a rubric may improve throughput and consistency, but may equally propagate its biases into judgements later reported as human. Whether assistance helps, harms, or is neutral is an empirical question current platforms cannot address, since assistance, where available at all, is neither constrained, nor logged, nor randomised.

We present **Aura**, a self-hostable web application addressing both problems. An administrator configures a *campaign* (a task family, an evaluation protocol, criteria, a scale, and a segment file containing the items to be judged; §3); annotators with portable accounts contribute judgements, linked across campaigns by a stable UUID; and Aura returns a reproducible results pack for each campaign or pooled *group*. None of these capabilities is new in isolation (§2); our contribution is to integrate them in one self-hostable system that, uniquely to our knowledge, treats assistant availability as an experimental variable. Specifically:

1. A **task- and protocol-general** platform: five protocols (Likert/DA, pairwise, ranking, error-span, post-editing) across eight task-family presets, sharing one segment schema and result model (§3).

2. A **grounded, advisory LLM assistant** that answers comprehension and guideline questions, prompt-constrained (not enforced) never to suggest a score, label, or span, with every interaction logged (§A).

3. A **built-in randomised experiment** that makes the assistant available on a random fraction of {annotator, segment} pairs and reports an intent-to-treat estimate of its effect on agreement and time-on-task (§A).

4. An **under-resourced-first** design: first-class script and speaker-provenance fields, language-agnostic metrics, no reference or paid crowd required, and offline single-machine operation; the seeded demo covers nine North-East Indian languages (§4).

5. An **agreement and diagnostics suite** (Krippendorff's α, Cohen's and Fleiss' κ, span-F_1, DA correlations; §5).

2 Related Systems

Table 1 situates Aura among widely used annotation and evaluation tools. MT-specific platforms such as Appraise (Federmann, 2018) and Anthea (Freitag et al., 2021) implement DA and MQM well but are tied to translation and to their own protocols. General-purpose tools like Label Studio (Tkachenko et al., 2020–2024), POTATO (Pei et al., 2022), INCEpTION (Klie et al., 2018), doccano (Nakayama et al., 2018), and BRAT (Stenetorp et al., 2012), are highly configurable, but oriented toward *labelling* rather than comparative with automatic evaluation (Singh et al., 2025), and do not always provide the MT/NLG result dashboards (span-F_1, per-criterion κ/α, win rates etc) that evaluation studies need.

For fine-grained work, Thresh (Heineman et al., 2023) unifies span evaluation but omits scoring and pairwise ranking while factgenie (Kasner et al., 2024) annotates spans in generated text and can substitute an LLM for the human annotator. Model assistance also appeared in INCEpTION (recommenders/ pre-labels) and AnnoHID (Ningtyas et al., 2026) (LLM pre-annotation with human validation), but in all of these the model supplies the *answers*.

Each capability in Table 1 is available somewhere: Appraise and Anthea cover DA and MQM for MT, Thresh and factgenie cover span evaluation, and INCEpTION, factgenie, and AnnoHID bring models into the loop. What Aura contributes is their combination in one system: DA, pairwise, ranking, MQM-style spans, and post-editing under one schema, with speaker and script provenance, agreement dashboards, and a reproducible results pack. Where prior tools use a model to supply answers, Aura's assistant only explains, and its availability can be randomised and logged. To our knowledge, this last capability has no counterpart in existing platforms.

3 System Overview

Roles and workflow: An *administrator* creates a campaign, uploads segments, and shares a per-campaign link. *Annotators* register once and reuse the same account across any campaign; a stable account UUID is the join key that lets a study span several campaigns (e.g. one per difficulty level or system) that Aura can later pool into a *group*. Ratings are saved server-side and remain editable until the administrator closes the campaign, at which

	Aura	Appraise	Anthea	Thresh	factgenie	Label Studio	POTATO	INCEpTION	doccano
Scoring / DA	✓	✓	–	–	∼	✓	∼	∼	–
Pairwise / ranking	✓	✓	–	–	–	∼	∼	–	–
Error-span (MQM-style)	✓	∼	✓	✓	✓	✓	∼	✓	∼
Post-editing	✓	∼	–	–	–	∼	–	–	–
Non-MT NLG tasks	✓	–	–	✓	✓	✓	✓	✓	✓
Built-in IAA dashboard	✓	∼	∼	–	∼	✓	✓	✓	–
Reproducible results pack	✓	∼	∼	✓	∼	∼	✓	✓	∼
Annotator qualification tests	✓	–	–	–	–	–	∼	–	–
Grounded LLM *assistant*	✓	–	–	–	–	–	∼	∼	–
Randomised AI-assist experiment	✓	–	–	–	–	–	–	–	–
Self-hostable / open source	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Feature comparison. ✓ = supported; ∼ = partial or possible with configuration/plugins; – = not supported. Entries for other tools reflect documented, publicly available functionality of current releases as of early 2026.

point Aura freezes the data and builds the results archive.

One schema, many protocols: A campaign's evaluation data is supplied as a segment file,[2] a single upload listing every item annotators will judge, one segment per record. Each segment carries an id, the source (input) text, and the output to be evaluated: target for single-output protocols, target_a/target_b for pairwise preference, or a candidates list for ranking. Optional reference, system, domain, and difficulty fields complete the record. Segment files may be JSON, JSONL, CSV, or TSV, with the format auto-detected. The same schema drives every protocol: *Likert/DA* (discrete 2-11-point scales rendered as circles, buttons, radios, or stars, or a continuous 0-100 slider), *pairwise preference* over an admin-defined option set, *ranking* of k candidates, *span-only* error annotation in which criteria become error-type categories, and *post-editing*. Criteria, scales, instructions, and even the protocol can be edited until closing; switching to a protocol the uploaded segments cannot support is rejected with an explanatory error. Figure 1 shows the annotator screen for a preference ranking campaign.

Annotator quality controls: Aura supports per-campaign *qualification tests* (gold segments with a pass threshold, retry limits, and cooldowns), optional background variables (native language, self-rated source/target fluency, dialect, location, and an age *range*), and a deterministic per-annotator segment shuffle that reduces ordering effects while being fully recorded (a *presentation_order.csv* maps each annotator's position→segment), so presentation order never silently affects results.

Workflow Overview: Aura is a self-hosted Flask web service[3] (Figure 2) that handles both roles: admins and annotators, persists to Postgres[4], calls a pluggable LLM provider through a logged, grounded gateway, and emits a reproducible results pack. All components except the optional LLM/email are local. In our workflow, admins create and configure multiple campaigns (e.g., evaluation protocol, annotation schema, qualification tests, and AI-assistant settings), each generating a unique shareable link for annotators. Annotators register or log in, complete their language profile, read task-specific instructions, and perform the assigned evaluation. Aura centrally manages authentication, annotator profiles, live analytics, quality monitoring, and response storage before exporting evaluation results.

The LLM layer is provider-agnostic (Anthropic, OpenAI, Google)[5]; per-campaign API keys are encrypted at rest with a key derived from the app secret. The only external dependency is an optional email transport used to deliver results; with none configured, everything remains downloadable. This makes Aura deployable on a single institutional VM with no cloud lock-in, which is important for data governance around unpublished system outputs and annotator data.

[2] E.g., a ranking segment in JSONL: `{"id": "s1", "source": "...", "candidates": [{"target": "...", "system": "A"}, ...]}`.

[3] Flask, https://flask.palletsprojects.com

[4] PostgreSQL, https://www.postgresql.org.

[5] Anthropic API, https://docs.anthropic.com; OpenAI API, https://platform.openai.com; Google Gemini API, https://ai.google.dev

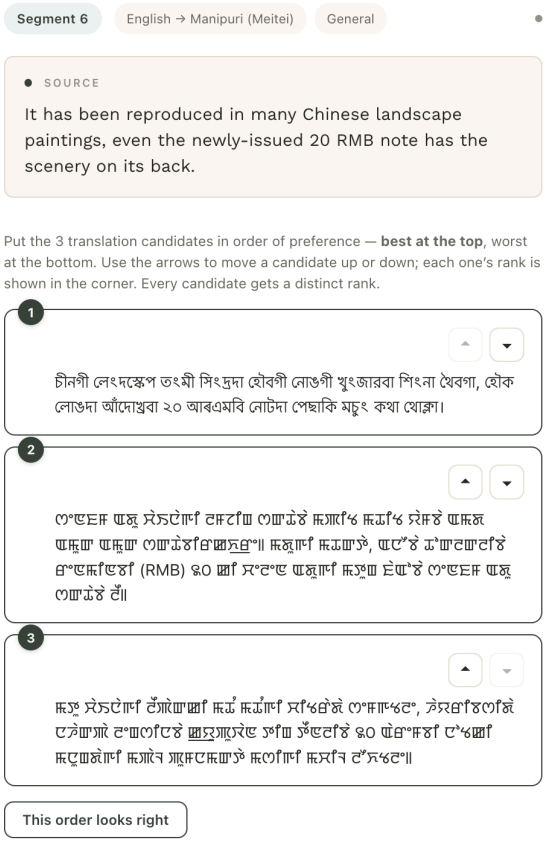


Figure 1: A snapshot of the annotation interface of AURA with collapsible rubric (English→Manipuri Preference Selection campaign).

4 Designed for Under-resourced and Indigenous Settings

Aura's defaults are chosen for evaluation of languages that lack large communities, paid crowds, or reference data, while remaining a general NLG tool.

Writing systems as first-class: Each campaign records script alongside language (e.g., *English (Latin) → Assamese (Bengali-Assamese)*), with digraphic languages such as Manipuri (Meitei Mayek) represented explicitly rather than collapsed. Character-level, Unicode-native span offsets keep error-span marking, span-F_1, and span density consistent across scripts.

Speakers, not interchangeable workers: Since judgements here depend on who is judging, Aura exports speaker provenance with every rating: native language, self-rated source and target fluency, dialect or variety, region, and coarse age range. This lets disagreement be traced to dialectal variation and annotators be stratified or weighted, rather than treated as a homogeneous pool.

No reference, no crowd, no cloud: Aura needs no human reference, and its metrics are language-agnostic: a lexicon-free, length-based readability proxy, and agreement computed from the ratings alone. It runs fully offline on one machine with Postgres, so unpublished outputs and community data need never leave a lab or a partner community's infrastructure (Nekoto et al., 2020; Bird, 2020).

Assistance where expertise is scarce: The grounded assistant (Appendix§A) helps most where annotators are few and multilingual, glossing terms or explaining rubric on the spot. Since LLM quality is weakest for these very languages, the randomised experiment lets a team *measure* whether it helps for their language rather than assume it.

5 Analytics and Reproducible Output

On close (or live, from the dashboard) Aura computes a protocol-aware analysis, following the standard definition of each statistic: per-criterion descriptives; Krippendorff's interval α (Krippendorff, 2004); Cohen's (Cohen, 1960) and Fleiss' (Fleiss, 1971) κ and percent agreement; pairwise win rates; character- and token-level span-F_1 with error-span *density* (mean span length and share of text flagged, so a one-word slip and a whole-clause error are not conflated); and DA correlations (Pearson/Spearman) between annotators. A *linguistic diagnosis* surfaces the most-contested segments with a language-agnostic readability proxy, a preview of annotator comments, and a drill-down to every rater's scores and spans, so a disagreement number links directly to the text that caused it. Results are delivered as a self-contained archive (*ratings.csv*, *results_summary.csv*, *results.json*, a standalone HTML report, and all figures as PNG) and can be emailed to the campaign owner. Group-level views pool compatible campaigns with n-weighted means while deliberately *not* pooling agreement across campaigns whose segments differ.

6 Evaluation

Deployment and user study: To demonstrate Aura in a real annotation setting, we deployed it to conduct human evaluation of English→Assamese, English→Manipuri, and English→Nagamese machine translation, representing three low-resource

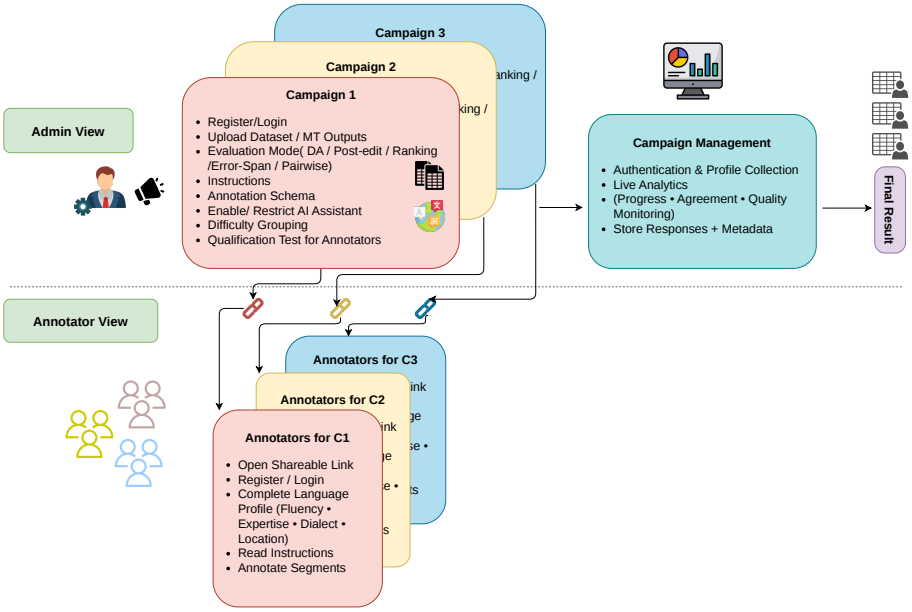


Figure 2: **Overview of Aura's workflow**. Matching colors associate Campaign 1 (**C1**) ( ), Campaign 2 (**C2**) ( ), and Campaign 3 (**C3**) ( ) with their corresponding annotator groups. DA = Direct Assessment.

language pairs from North-East India. The evaluation compared translation outputs generated by three large language models (LLMs): *Gemini-2.5-flash-lite*, *Claude-sonnet-4.6*, and *GPT-4.1-mini*. The study involved nine annotators ($N = 9$), with each language pair evaluated independently by three native or proficient speakers of the target language. For each language pair, annotators were presented with 30 source segments, each accompanied by translation outputs from the three LLMs. Their task was to rank the candidate translations from best to worst according to their overall translation quality. Aura automatically generated reproducible evaluation reports, including pairwise ranking results, inter-annotator agreement statistics, annotator performance summaries, and linguistic diagnostics for contested segments, together with downloadable result archives for each campaign (Table 2). These results illustrate Aura's output rather than constitute a model comparison: with 30 segments and three annotators per language pair, differences in Elo and win rate are not statistically robust. We did not perform bootstrap or resampling-based stability analysis; thus, the Elo and win-rate values are treated as illustrative. Annotators used portable accounts across campaigns and received an honorarium of INR 500 for completing the study. This study exercises one protocol (ranking) on one task family (MT). Likert/DA, pairwise, error-span, and post-editing, and the non-MT task families, are implemented and included in the seeded demo, but were not evaluated with users here.

User-experience and AI-assistance evaluation: Participants completed the standard ten-item System Usability Scale (SUS) questionnaire (Brooke, 1996) together with free-text feedback after annotation. In addition to the nine annotators in the initial ranking study, three additional participants, all co-authors of this paper, evaluated the platform on English→Manipuri with the advisory AI assistant enabled, resulting in 12 participants for usability assessment. No reference translations were used in any campaign, so the assistant had no reference to disclose.

Across the AI-assisted English→Manipuri campaign (30 source segments, 90 candidate rankings), all three raters utilized the assistant, logging 11 interactions across 5 distinct ratings. Mean time-on-task was substantially higher for assisted ratings than unassisted ones (**1454.8** s vs. **1134.8** s). Annotator telemetry indicates that raters did not rely on the assistant as an automated shortcut; rather, queries were selectively triggered during deliberation on difficult, highly contested segments (e.g., lexical ambiguity and severe translation errors). Assisted annotations retained solid inter-rater consistency (68.15% pairwise agreement; mean Kendall's $\tau = 0.363$), comparable to unassisted Manipuri evaluation (69.44%; $\tau = 0.389$), demonstrating that advisory assistance preserves rater autonomy and label integrity.

Measure	Eng→Asm	Eng→Nag	Eng→Mni	Eng→Mni (+AI)
Annotators	3	3	3	3
Source segments	30	30	30	30
Candidate LLMs	3	3	3	3
Pairwise agreement (%)	65.56	77.78	69.44	68.15
Mean Kendall's τ	0.311	0.556	0.389	0.363
Top-ranked LLM	Gemini	GPT	Gemini	Gemini
Elo rating	1105.3	1030.1	1218.8	1119.9
Win rate (%)	68.0	63.3	81.7	73.9
AI interactions	–	–	–	11
AI-assisted ratings	–	–	–	5 / 90
Median time per segment (s)	–	–	1134.8	1454.8

Table 2: Summary of ranking-based human evaluation across the three language pairs, including the AI-assisted English→Manipuri campaign.

Across all 12 participants, Aura achieved a mean SUS score of **82.29**/100 (SD = **16.53**, $N = 12$), indicating good perceived usability. Feedback highlighted three main themes: (i) evaluation-dataset quality, particularly unnatural or orthographically inconsistent translations in Nagamese and Manipuri; (ii) minor UI/UX issues, including rendering, navigation, and page-transition delays; and (iii) positive perceptions of the interface and AI assistant, particularly its usefulness for understanding evaluation context and requirements. Annotators otherwise appreciated the simplicity of the annotation workflow. The SUS questionnaire combines positively and negatively worded items, with alternating polarity designed to encourage respondents to consider each statement carefully and reduce response bias.

7 Audience, Impact, and Licensing

Audience: Aura targets researchers, community organisations, and language activists evaluating NLG for *indigenous and low-resource languages*, who need rigorous human evaluation without a paid crowd, a reference, or moving data off community control (Joshi et al., 2020; Nekoto et al., 2020); the seeded demo covers nine North-East Indian languages. It equally serves MT and NLG researchers, shared-task organisers, and industry teams comparing model versions, the "and beyond" of our title.

Impact: Implementing five protocols and eight task families in one reproducible tool lowers the fixed cost of human evaluation; for under-served languages, removing the paid-crowd and cloud requirements can decide whether a careful evaluation happens at all. Making AI assistance a logged, randomisable variable offers an off-the-shelf way to study a practice spreading faster than our understanding of its effects on label quality.

Licensing: Aura is released under the MIT License, permitting commercial and academic use, modification, and redistribution with the copyright notice retained; third-party dependencies keep their own open-source licenses.

8 Conclusion

Aura implements five human-evaluation protocols for NLG in a single open-source, self-hostable platform designed for under-resourced languages, with script and speaker provenance, agreement analytics, and reproducible results archives. It adds a grounded, advisory, and logged LLM assistant whose availability can be randomised, making the effect of AI-in-the-loop annotation measurable rather than assumed. A pilot on three North-East Indian MT ranking campaigns demonstrates the ranking protocol; validating the remaining protocols and running the randomised assistance experiment are future work. and its availability can be randomised, making the effect of AI-in-the-loop annotation measurable rather than assumed.

Limitations

Aura's A/B estimate is intent-to-treat: it measures the effect of the assistant's availability, not its use. Deviation from consensus rewards majority agreement, not correctness. The assistant's advisory role is enforced only by its prompt. Aura's Elo depends on comparison order; Bradley-Terry with bootstrap intervals suits small studies better. Only ranking on MT was tested with users. The AI-assisted pilot (§6) is observational, with low uptake (5 of 90 ratings), so it cannot estimate the assistant's effect.

Ethics and Broader Impact

Aura collects annotator emails and optional background variables (native language, fluency, dialect, location, age *range*) with informed consent, kept under the deployer's control. We paid the nine independent annotators INR 500 each and encourage fair compensation for speakers of under-served languages. For these languages, a few speakers' judgements can be identifying; Aura's offline design keeps data under community control but does not confer consent, so deployers should engage communities as partners (Nekoto et al., 2020; Bird, 2020). An assistant can pull labels toward the model's own distribution; Aura logs every interaction, but its constraints are prompt-only, so assisted labels should not be reported as assistance-free.

References

Steven Bird. 2020. Decolonising speech and language technology. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3504–3519, Barcelona, Spain (Online). Association for Computational Linguistics.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

John Brooke. 1996. SUS: A 'quick and dirty' usability scale. In Patrick W. Jordan, Bruce Thomas, Bernard A. Weerdmeester, and Ian L. McClelland, editors, *Usability Evaluation in Industry*, pages 189–194. Taylor & Francis.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Jay Gala, Pranjal A. Chitale, A. K. Raghavan, Sumanth Doddapaneni, Varun Gumma, Aswanth M. Kumar, Janki Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M. Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. IndicTrans2: Towards high-quality and accessible machine translation models for all 22 scheduled Indian languages. *Transactions on Machine Learning Research*.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30).

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

David Heineman, Yao Dou, and Wei Xu. 2023. Thresh: A unified, customizable and deployable platform for fine-grained text evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 336–345, Singapore. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. 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 and 1 others. 2024. factgenie: A framework for span-based evaluation of generated texts. In *Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations*.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tom Kocmi, Vilém Zouhar, Eleftherios Avramidis, Roman Grundkiewicz, Marzena Karpinska, Maja Popović, and Jason Riesa. 2024. Error span annotation: A balanced approach for human evaluation of machine translation. In *Proceedings of the Ninth Conference on Machine Translation (WMT)*, Miami, Florida, USA. Association for Computational Linguistics.

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*, 2nd edition. Sage Publications.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human.

Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, and 1 others. 2020. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160, Online. Association for Computational Linguistics.

Annisa Maulida Ningtyas, Guntur Budi Herwanto, Yunita Sari, Rifki Afina Putri, Filip Kovacevic, Alaa El-Ebshihy, Varvara Arzt, and Florina Piroi. 2026. AnnoHID: LLM-assisted annotation framework for low-resource medical texts. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 683–691, San Diego, California, United States. Association for Computational Linguistics.

Jiaxin Pei, Aparna Ananthasubramaniam, Xingyao Wang, Naitian Zhou, Apostolos Dedeloudis, Jackson Sargent, and David Jurgens. 2022. POTATO: The portable text annotation tool. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 327–337, Abu Dhabi, UAE. Association for Computational Linguistics.

Kshetrimayum Boynao Singh, Deepak Kumar, and Asif Ekbal. 2025. Evaluation of LLM for English to Hindi legal domain machine translation systems. In *Proceedings of the Tenth Conference on Machine Translation*, pages 823–833, Suzhou, China. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at EACL 2012*.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020–2024. Label Studio: Data labeling software.

Xinru Wang, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao. 2024. Human-LLM collaborative annotation through effective verification of LLM labels. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*.

A Details on the Grounded Assistant

Grounded AI assistant: Aura provides a dockable assistant answers annotators' free-form questions, restricted to two functions: explaining source or generated text (vocabulary, grammar, style, translation), and clarifying guidelines, criteria, and interface behavior from the campaign configuration. Its system prompt instructs it never to recommend labels, preferences, or spans, nor to judge system outputs, and not to disclose the hidden reference. These are prompt-level constraints: Aura does not validate the model's output, so compliance depends on the underlying LLM, and every exchange is logged so violations can be detected after the fact. All interactions are logged in *assistant_log* with annotator, segment, and action metadata.

Evaluating AI assistance: Aura supports both observational and randomized evaluations of AI assistance. Observationally, it compares assisted and unassisted annotations via logged interactions. To estimate causal effects, Aura randomizes assistant availability across (annotator, segment) pairs (A/B), yielding an intent-to-treat estimate: it reports mean deviation from consensus and annotation time for both groups, quantifying the impact of AI assistance on annotation quality and efficiency.