Human + AI: Large scale Data Curation For Multilingual Guardrails

Harshit Rajgarhia
Centific
harshit.rajgarhia@centific.com

Abhishek Mukherji
Centific
abhishek.mukherji@centific.com

Fen Yik
Centific
fen.yik@centific.com

Dominika Borek
Centific
dominika.borek@centific.com

Nicole Warren
Centific
nicole.warren@centific.com

Prithiviraj Pradeep
Centific
prithiviraj.pradeep@centific.com

ABSTRACT

As Large Language Models (LLMs) become increasingly central to real-world applications, the demand for high-quality, instruction-compliant, and multilingual training data has surged, particularly in tier-2 languages with limited digital representation. In this work, we introduce an AI-assisted annotation framework designed to optimize authoring of training data for multilingual guardrails, specifically PII detection, in Supervised Fine-Tuning (SFT) of LLMs. Targeting 13 locales, mostly underrepresented, we operationalize a suite of AI tools to augment human annotators without replacing them. Our results demonstrate a 40+% reduction in average handling time while improving instruction compliance, semantic diversity, and data quality. The key contribution of this work is that we explore the emerging paradigm of 'LLM-as-a-Judge', using LLM not only as generative tools but also as evaluators of human-authored training data.

VLDB Workshop Reference Format:
Harshit Rajgarhia, Abhishek Mukherji, Fen Yik, Dominika Borek, Nicole Warren, and Prithiviraj Pradeep. Human + AI: Large scale Data Curation For Multilingual Guardrails. VLDB 2025 Workshop: DaSH:Data Science with Human in the Loop.

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at URL_TO_YOUR_ARTIFACTS.

1 INTRODUCTION

In order to improve the efficacy of large language models, the foundational AI companies are rapidly consuming much of the digital content in all modalities. To that end, the Epoch AI blog [13] predicts that much of this digital content will be fully consumed for training the LLMs between 2026 and 2032, yet there will still be scope for LLMs to be effective in solving several domain and industry-related real-world problems. Thus, the rapid emergence of data curation and annotation industry particularly focused on human-led instruction-compliant data curation. Several commercial data annotation platforms and services have emerged as stated in this Blog [2] to meet the demand. Human-led data annotation and synthesis generally refers to the labeling or generation of raw data with task-specific instructions or guidelines compliance.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment. ISSN 2150-8097.

1.1 The Annotation Workflow

The annotation process consists of *people*, *process* and *technology*.

The People: Each of these companies have a global workforce of quality managers, linguists, and native speakers. While they bring the language expertise, the data curation for task-specific LLM training requires expertise in understanding specifics such as the domain (legal, tech, finance, etc.) as well as collection of accurate metadata to enable Supervised Fine-Tuning.

The Process: The data curation process is labor intensive and costly. The workforce of native speakers and linguists from the crowd forms the group of annotators. At times, each data curation task is assigned to single or multiple annotators. The annotated data are then reviewed by a more experienced user on the task, such as the Quality manager. These expert reviewers are more expensive resources than annotators. Hence, depending on the complexity of the task, the number of annotators and reviewers is in the 2:1 or 3:1 ratio. A common practice requires multiple, typically 2, annotation tasks must be compared and agreed upon, only if the annotators disagree on the task below a certain threshold, a third arbitrator selects the correct answer as-is or with minor fixes. The tasks rejected by the reviewers are passed back into the pool of annotators. We monitored annotators' performance (common error categories and acceptance/rejection rates), as well as reviewers' performance (accuracy and efficiency on tasks), to manage the resource pool and maintain a high level of data quality.

The Technology: The past few years have seen the emergence of several data annotation platforms. Traditionally, such data curation is done on Excel spreadsheets. More recently, several data annotation platforms [5, 27] have emerged that have SaaS cloud or on-prem (for data privacy) offerings. Such annotation platforms provide features such as SSO access, several reusable templates for multi-modal data curation, standard or custom agreement metrics, project configurations and progress tracking, and quality dashboards.

A typical annotation workflow as shown in Figure 1 begins with an annotator from a pool of annotators performing a task based on the given set of meta-data instructions. Once the task is submitted, a reviewer (from a pool) picks the task to audit it and then takes one of the following actions:

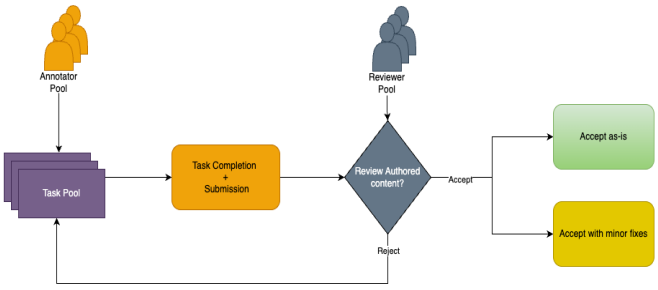

Figure 1: Annotation Workflow

- Accepts the task as-is.
- Makes minor fixes and then accepts the task.
- Rejects the task, returning the task to the original pool of tasks to be picked up by another annotator.

The reviewer's decision is based on a set of explicit quality criteria or instruction, such as the `WORD COUNT`, as well as implicit quality criteria, such as whether the language is appropriate for the specified locale.

Annotation Performance: We define *handling time* as the total time required to complete a task (annotation + review) that meets all necessary quality standards. The success of an AI-assisted annotation workflow is determined by minimizing Average Handling Time(AHT) while maximizing the quality of the completed task.

1.2 Challenges in Data Curation for LLM SFT

In the space of data curation for Supervised Fine-Tuning (SFT) of LLMs, below are the set of challenges that we address in this work.

(1) While current LLMs continue to become more effective in English and a few handful of global languages, LLMs still lack basic understanding of tier-2 underrepresented languages listed in Table 1. Thus, expansion to these languages requires dedicated efforts in data curation.
(2) Further, detection of PIIs or guardrails in these languages is difficult due to limited digital representation. These PIIs are country-specific and, while, certain PII types such as `DRIVER'S LICENSE`, `SSN`, `BANK ROUTING NUMBER`, `PASSPORT` have well-defined formats, other PII types such as `NAME` and `ADDRESS` are open-ended. Use of Real PIIs for LLM training poses security risks; manually generating synthetic PIIs, that resemble real PIIs, is time consuming.
(3) While traditionally the linguists or Quality managers are skilled at language translation tasks, additionally, data curation for LLM SFT must be compliant with specific guidelines or instructions and affix accurate metadata (such as TASK CATEGORY, DOMAIN and INTENT).
(4) The annotators often adopt a formulaic approach to writing style or repeatedly use certain words and phrases, which may hamper the lexical and syntactic diversity in the authored content.
(5) As data curation for LLM SFT is emerging, the guidelines are often ambiguous and the application of instructions must be clarified by annotators and / or reviewers as part of the process. This leads to quality challenges as well as much rework, which, in turn, adds to the high AHT.

Locale	Prompt Volumes
ar-UAE	5000+
fi-FI	5000+
hi-IN	5500+
nb-NO	5000+
nl-BE	1000+
nl-NL	3500+
pl-PL	5000+
pt-BR	3000+
pt-PT	1500+
sv-SE	5000+
vi-VN	4000+
zh-CN	3000+
zh-SG	1000+

Table 1: Locale-wise prompt volumes.

1.3 Key Contributions

The key contributions of this work is as follows:

(1) In this work, we focus on authoring prompts as the annotation task in 13 locales. In addition to the human aspects of people and process, this paper focuses on the technological advances that helped with expediting **authoring 40000+ prompts across 13 locales, mostly underrepresented**.
(2) This work specifically addresses multi-lingual guardrails, i.e., authoring prompts that contain PII information that resemble real PII (such as `PASSPORT NUMBER`, `BANK ACCOUNT NUMBER`, `HEALTH ID`, `DRIVER'S LICENSE`) in the countries where these languages are spoken. To that end, our solution provides live PII suggestions to annotators based on the PII categories required per task. **Overall, we cover 300+ PII types across the 13 locales**.
(3) The key contribution of this work is a set of AI-Assisted Instruction Following Validations that provides proactive guidance to both annotators and reviewers. LLM-as-a-judge is an emerging research topic and **our work is the first to leverage LLMs to judge human annotations**.
(4) The combination of PII suggestions, the instruction-following and the near duplicate check contributed to **reduced AHT by 40+%, superior data quality and diversity**.

2 RELATED WORK

This section is primarily informed by the comprehensive survey by Tan et al. [29], which offers an in-depth taxonomy and review of LLM-based data annotation. AI-assisted annotation has gained significant traction in recent years, driven by the increasing scale of data and the limitations of manual labeling. The emergence of Large Language Models (LLMs), such as GPT-4 [24], Gemini [30], and LLaMA-2 [32], has opened new avenues for automating data annotation with high accuracy and contextual sensitivity. These models serve not only as automatic annotators but also as tools for

data augmentation, quality assessment, and feedback generation [29].

LLM-based annotation can take several forms, including prompt-response generation [25], label generation [36], rationale generation via chain-of-thought prompting [17], and pairwise feedback synthesis for preference learning [4]. These methodologies significantly enhance the diversity, scale, and quality of annotated datasets, providing valuable resources for downstream tasks such as supervised fine-tuning and alignment tuning [6, 20]. In this work, we focus on the prompt authoring for training LLMs for multi-lingual guardrails.

In addition to annotation generation, LLMs can assess and filter annotations using rule-based heuristics [35], pretrained reward models [12], or LLM-based evaluation mechanisms such as LLM-as-a-Judge [37]. These approaches ensure high-quality data selection, essential for preventing label noise in large synthetic datasets.

Several studies have explored the integration of LLM-generated annotations into different training strategies, such as self-training, in-context learning, and instruction tuning. For instance, Huang et al. [16] and Yang et al. [38] explore iterative fine-tuning and self-distillation to enable LLMs to refine their own outputs. Such methods contribute to more robust and generalizable models, even in low-resource settings.

Beyond LLM-exclusive paradigms, hybrid approaches such as Human-in-the-Loop (HITL) annotation frameworks are gaining traction. These systems use LLMs to pre-annotate data, followed by human validation to improve accuracy, especially in edge cases [18]. Similarly, weak supervision and data programming allow developers to use labeling functions, ontologies, or external sources to annotate data, often refined with LLM feedback to enhance reliability and reduce redundancy.

Domain-specific annotation remains an active area of research. LLMs have been applied to automate annotation in sensitive fields such as medicine [15], finance [19], and legal reasoning [7], where precision and ethical considerations are paramount. This has driven interest in instruction tuning with domain-relevant prompts, as well as collaborative annotation pipelines that blend expert knowledge with LLM assistance.

Despite these advancements, key challenges remain. Hallucinations and sampling bias are ongoing concerns when LLMs generate unreliable or unverifiable annotations [3]. Furthermore, model collapse [28]—where iterative training on synthetic LLM data leads to performance degradation—raises questions about the long-term viability of purely synthetic annotation pipelines. Ethical dilemmas, such as embedded bias [1] and workforce displacement [11], also warrant caution.

While the use of LLMs for data generation and augmentation has been widely studied, research on leveraging LLMs as judges or evaluators of human-authored annotations remains relatively limited. Only a few recent efforts have begun to investigate the reliability, consistency, and potential biases of LLMs in this evaluative role. Our work is among the first to operationalize this paradigm in a production-scale multilingual annotation pipeline, using LLMs not only to assist with instruction-following but also to proactively assess the quality of prompts authored by human annotators. This positions our approach at the frontier of AI-assisted quality assurance in human-in-the-loop workflows.

PII Types		
ADDRESS	AGE	AWS ACCESS KEY ID
AWS SECRET KEY	BANK ACCOUNT NUMBER*	BANK ROUTING*
CREDIT DEBIT CVV	CREDIT DEBIT EXPIRY	CREDIT DEBIT NUMBER
DATE	DRIVER ID*	EMAIL
HEALTH ID*	IP ADDRESS	LICENSE PLATE*
MAC ADDRESS	NAME	NATIONAL ID*
PASSPORT NUMBER*	PASSWORD	PHONE
PIN	SSN*	SWIFT CODE
TIN*	URL	USERNAME

Table 2: List of PII types used in the study.

3 THE PROMPT AUTHORING TASK

The annotation task at hand involves generating prompts that contain synthetic Personally Identifiable Information (PII) to simulate interactions with Large Language Models (LLMs). PII refers to any information that can directly or indirectly identify a specific individual. The categories of PII may be general or specific to a particular locale or domain.

This task encompasses 13 locales as mentioned in Table 1. In addition to incorporating PII, each prompt must be generated in compliance with the metadata instructions outlined below:

(1) `CATEGORY`: Each prompt must belong to exactly one of the following task types: Generate, Q&A, Classification, Rewrite, Conversation, Summarize, Translation,Extraction, or Chain-of-Thought.
(2) `DOMAIN`: Each prompt must be related to exactly one of the following domains: Finance, Travel, Healthcare, IT, CPG, or Media.
(3) `PII ENTITIES`: Each prompt must contain at least one synthetic PII for each PII entity provided as input to the task. Table 2 mentions the different types of PII used in the study. PII types that are marked with * indicates that these PII types are different for each locale. For example, `AADHAR ID` is used in hindi locale while `NATIONAL ID` is used in Arabic, Norwegian, Polish, etc. `SSN` (Social Security Number) is used for Norwegian, Polish, Dutch while `PAN` (Permanent Account Number) is used in Hindi.
(4) `WORD COUNT`: Each prompt must fall within the specified length categories:
    - Small (<30 words)
    - Medium (30–200 words)
    - Large (200–1000 words)
    - Extra-Large (1000–3000 words)
(5) `DISCLOSURE TYPE`: Each prompt must specify either an implicit or explicit information type.
    - An "explicit" request directly states that the provided information corresponds to a relevant PII category.
    - An "implicit" request does not explicitly mention that the information belongs to a PII category.
(6) `REQUEST TYPE`: Each prompt must be categorized as either structured or unstructured.
    - A "structured" request explicitly requires the response to be formatted in JSON.
    - An "unstructured" request does not specify any required response format.

(7) INTENT: Each prompt must be categorized as either malicious or non-malicious.
   - A "malicious" use case involves crafting prompts where the intent behind using PII is harmful or unethical.
   - A "non-malicious" use case involves prompts where the intent behind using PII is legitimate and ethical.

4 AI-ASSISTED ANNOTATION SOLUTIONS

We designed and put in production a suite of AI enabled tools to assist the annotators to author prompts containing PII information. The motivation behind development of these solutions were not to auto-annotate or replace the human layer of annotation, but to rather assist the Human-in-the-loop. The efficacy of these solutions was evaluated based on the ability to decrease the overall Average Handling time and to produce higher Quality annotations.

4.1 Synthetic PII Generator

This module was developed to reduce Average Handling time, as discussed in subsection 1.2. In total, there are 300+ distinct PII types spread across 13 locales. While some PIIs are locale-agnostic, such as DATE, a significant portion are locale-specific, such as LICENSE PLATE.

To address this, Python-based Synthetic data generation libraries such as Faker[14] and Mimesis[26] were leveraged. Using regular expressions and custom rules, these libraries were extended by creating custom providers that adhere to locale-specific guidelines and formats. Additionally, a data bank of 6000 synthetic PIIs was built for each PII entity in each locale, totaling 2M synthetic PIIs.

The Synthetic PII Generator was deployed within the annotator workflow, where it pre-populates suggested PIIs for annotators at the task level. To eliminate duplication, the generator was designed to ensure that an entity is not suggested to an annotator if it has already been used in a prompt by another annotator. This solution significantly reduced annotators' workload, eliminating the need to manually search the web and craft synthetic PIIs that comply with locale-specific rules.

4.2 Instruction Following

This module was developed to enhance annotation quality and reduce Average Handling time, as discussed in subsection 1.2. As outlined in section 3, each prompt authoring task includes a set of meta-data instructions[CATEGORY, DOMAIN, WORD COUNT, DISCLOSURE TYPE, REQUEST TYPE and INTENT] that must be followed during prompt creation. Failure to adhere to these instructions results in repeated interactions between the annotator and the reviewer, leading to an increase in Average Handling time. Based on the approach taken to validate if the authored prompt complies with the meta-data instructions, this module was divided into three sub-sections

- CATEGORY, DOMAIN, DISCLOSURE TYPE, REQUEST TYPE and INTENT: To ensure compliance with these instructions, few-shot Large Language Models (LLMs) are utilized. We experimented with several LLM models, including GPT-3.5[22], GPT-4o[23], and LLaMA 2[32], and found that GPT-4o outperformed the others in terms of performance. This approach is supported by documented cases demonstrating that LLMs outperform other language models and traditional machine learning models in various open-source tasks. Beyond improved performance, zero-shot LLMs offer the advantage of not relying on supervised labeled data, enhancing the adaptability of the instruction-following methodology in scenarios where obtaining labeled data is challenging or impractical.
  For each meta-data instruction type, a custom instruction-checking prompt is crafted to verify whether the corresponding instruction has been followed in the authored prompt. To fine-tune and refine the instruction-checking prompt, a representative supervised dataset generated by SMEs was utilized and the results were benchmarked as shown in Figure 2 . To improve responses and assist human(s)-in-the-loop, this module also generates reasoning to explain whether an instruction has been adhered to.
- WORD COUNT: A multilingual Word Count feature has been developed to calculate the number of words in a given text while accounting for the text's locale. This feature supports multiple languages, including non-Latin scripts such as Arabic, Hindi, and Chinese, as well as multilingual texts.
  The text is tokenized into words using regular expressions specifically designed for different scripts. Regular expressions are defined to match words, and each identified word is added to a list. To handle non-Latin scripts, special character classes such as \p{Devanagari} for Hindi, \p{Arabic} for Arabic, and appropriate patterns for Chinese are incorporated. Latin-based languages utilize a general regex pattern, while Hindi, Arabic, and Chinese employ tailored regex patterns suited to their respective script characteristics. The appropriate regex is selected based on the specified locale.
  For Arabic text, additional processing is applied using libraries such as arabic reshaper and bidi.algorithm[8] to ensure proper text rendering. Finally, the total word count is determined by calculating the length of the extracted word list.
- Language Detection: This module was developed to prevent annotators from authoring prompts in a language other than the desired one. Various Python libraries, such as langid[10] and langdetect[9], along with cloud-based

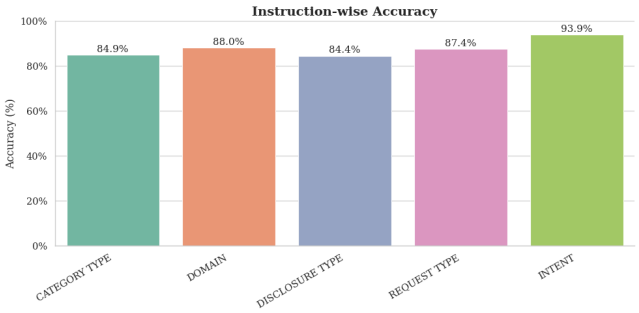

Figure 2: Accuracy of Instruction Following

Table 3: Overview of AI-Assisted Annotation Components

Component	Solution Approach	Impact
`Synthetic PII Generator`	Generates synthetic PII using libraries like `Faker` and `Mimesis`, extended with locale-specific rules. Prevents duplication across annotators.	Reduces manual effort and AHT while adding diversity.
`Instruction Following`	Uses few-shots LLMs to verify prompt compliance with metadata fields.	Reduces review iterations and improves quality.
`Word Count`	Tokenizes text using locale-specific regex. Supports Latin scripts and non-Latin scripts like Arabic, Hindi, and Chinese as well.	Ensures compliance with multilingual length limits.
`Language Detection`	Detects prompt language using libraries like `langid` and cloud services like `Azure Language Service`.	Prevents language mismatches across locales.
`Near Duplicate Check`	Computes semantic similarity using pretrained embeddings. Flags similar prompts to encourage variation.	Promotes prompt diversity and reduces redundancy.

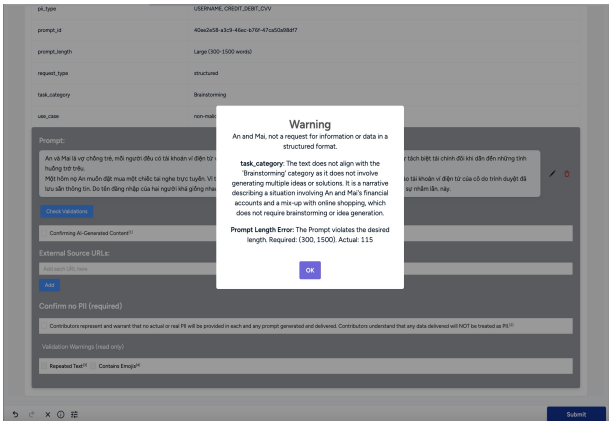

Figure 3: AI-assisted Annotation Platform

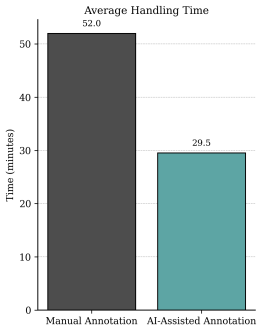

Figure 4: Comparison of AHT across all tasks and locales between manual and AI-assisted annotation.

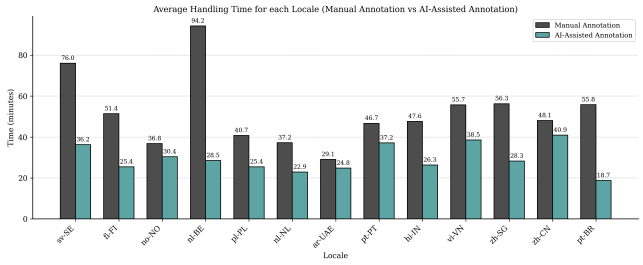

Figure 5: AHT comparison across all tasks at locale level between manual and AI-assisted annotation.

services like `Azure Language Service`[21], were evaluated. Among these, `Azure Language Service` was found to be the most effective in detecting the correct language across the 13 locales.

4.3 Near Duplicate Check

To maintain diversity within the prompts authored by an annotator, a semantic similarity detection module is employed. This module leverages various pretrained word-embedding models[31, 33, 34] based on the input language[1] and utilizes cosine similarity to generate a similarity score, $S_{\cos}$, ranging from 0 to 1. The cosine similarity between two vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ is defined as:

$$S_{\cos}(\mathbf{v_1}, \mathbf{v_2}) = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\|\|\mathbf{v_2}\|}$$

where $\mathbf{v_1} \cdot \mathbf{v_2}$ is the dot product of the vectors, and $\|\mathbf{v_1}\|$ and $\|\mathbf{v_2}\|$ are the magnitudes of the vectors.

The module is deployed in a manner that extracts the last n prompts authored by a given annotator and calculates the pairwise semantic similarity score within this set. The pairwise similarity score S_{pairwise} for two prompts P_i and P_j is calculated as:

$$S_{\text{pairwise}}(P_i, P_j) = S_{\cos}(\mathbf{v_i}, \mathbf{v_j})$$

If any pair exceeds a predefined threshold t, where $t \in [0, 1]$, a warning message is displayed to the annotator, prompting edits to enhance semantic diversity. Specifically, if:

$$S_{\text{pairwise}}(P_i, P_j) > t$$

then a warning is triggered. Upon various experiments involving language experts, n was set to 3 and t was set to 0.85.

Figure 3 shows our AI-assisted Annotation Platform.

[1]
- `uer/sbert-base-chinese-nli` for Chinese
- `sentence-transformers/LaBSE` for Norwegian
- `sentence-transformers/paraphrase-multilingual-mpnet-base-v2` for all other supported languages.

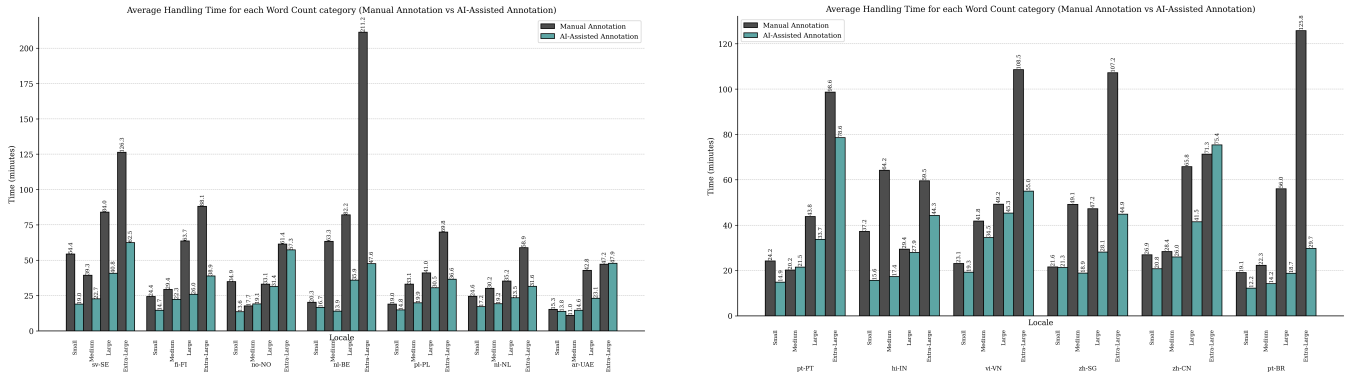

Figure 6: AHT comparison across all tasks at locale-Prompt Length level between manual and AI-assisted annotation.

5 RESULTS AND ANALYSIS

Average Handling time is a critical metric for assessing the efficacy of the AI-assisted annotation workflow. To analyze Average Handling time, an experimental setup was designed with two workflows: one being the Manual Annotation Workflow and the other the AI-assisted Annotation Workflow. In both workflows, 100 prompt authoring tasks per locale were distributed to two separate groups, each comprising the same number of annotators and reviewers. The distribution of prompt authoring tasks was identical in terms of meta-data instructions across both Workflows. The *Handling time* was collected at a task level and aggregated at various levels to draw insights.

It can be observed in Figure 4 that the Average Handling time decreases significantly(~ 43%) in the AI-Assisted Annotation Workflow compared to the Manual Annotation Workflow. Additionally, as shown in Figure 5, the Average Handling time is considerably lower (ranging from 15% to 70%) in the AI-Assisted Annotation across all locales when compared to the manual workflow.

A comparison of Average Handling time was also analyzed at the locale-`WORD COUNT` level. As shown in Figure 6, the Average Handling time is consistently lower in the AI-Assisted Annotation Workflow. Notably, for tasks with *extra-large* `PROMPT LENGTH`, the Average Handling time is significantly lower compared to other `PROMPT LENGTH` variants.

6 FUTURE WORK

While our AI-assisted framework has demonstrated significant improvements in multilingual prompt authoring for guardrails, several opportunities remain as discussed below.

(1) Quantitatively isolate and measure the impact of each AI-assisted component on both AHT and annotation quality.
(2) Additionally, although this paper focuses on the upstream task of data curation and workflow optimization, the ultimate objective is to improve model performance downstream. In future work, we will report on the accuracy and robustness of LLMs fine-tuned using the curated multilingual dataset developed through our framework. This will include benchmarking against existing datasets, evaluating gains in low-resource language comprehension, and assessing improvements in guardrail adherence for sensitive PII-related tasks.
(3) Another promising direction is the incorporation of adaptive, feedback-driven annotation workflows by introducing mechanisms for real-time learning from annotator corrections and reviewer insights.
(4) Lastly, the role of LLM as an evaluator warrants deeper exploration. Towards that end, ensuring fairness, interpretability, and bias mitigation in LLM-based judgments remains an open challenge to be solved with explainability frameworks and human feedback to build trust in LLM-as-a-judge paradigm.

7 CONCLUSION

We introduce a novel AI-assisted annotation framework which is used to author 40000+ prompts across 13 locales. Via A/B testing, we demonstrate a substantial reduction in AHT, exceeding 40 + %, while improving instruction compliance and semantic diversity. Notably, our use of LLMs as evaluators within the human-in-the-loop highlights their emerging role not only as content generators but also as effective arbiters of quality in annotation pipelines.

As the demand for high-quality, locale-sensitive training data grows, our approach offers a scalable, efficient, and ethically robust solution. The techniques presented here are broadly applicable and can be extended to additional languages, annotation formats, and domains—contributing meaningfully to the evolving ecosystem of LLM training and evaluation.

ACKNOWLEDGMENTS

We would like to thank our Quality services team: Lokesh Rachuri, Akhil Pothanapalli, Nali Venkata Triveni, Vamshi Kirshna Pinni, Asif Shaik, Ajay Krishna Anugu, Santhoshraj Y, Mahesh Samba, Manoj Kumar Reddy Thumu, Nithya Tanvy Nishitha Sanka, G Praveen Kumar for helping us implement the AI assisted annotation services. Thanks to Centific QM team: Giovanna Conte, Joanna Skubisz, Rong Yin and Reese Tateo for helping us test and improve the AI-assisted solutions.

REFERENCES

[1] Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 298–306.

[2] Basic AI. [n.d.]. *Top 10 Best Data Annotation tools 2024.* Retrieved May 15, 2025 from https://www.basic.ai/blog-post/top-10-best-data-annotation-data-labeling-tools-2024

[3] Hussam Alkaissi and Samy I McFarlane. 2023. Artificial hallucinations in ChatGPT: implications in scientific writing. *Cureus* 15, 2 (2023).

[4] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073* (2022).

[5] Centific. [n.d.]. *Frontier AI data foundry.* Retrieved May 9, 2025 from https://centific.com/frontier-ai-data-foundry-wb

[6] Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)* 2, 3 (2023), 6.

[7] Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2023. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. *arXiv preprint arXiv:2306.16092* (2023).

[8] Bidi Algorithm Developers. 2020. bidi.algorithm. https://python-bidi.readthedocs.io/en/latest/ Accessed: 2025-05-07.

[9] LangDetect Developers. 2021. langdetect. https://pypi.org/project/langdetect/ Accessed: 2025-05-07.

[10] Langid.py Developers. 2021. langid.py. https://github.com/saffsd/langid.py Accessed: 2025-05-07.

[11] Danica Dillion, Niket Tandon, Yuling Gu, and Kurt Gray. 2023. Can AI language models replace human participants? *Trends in Cognitive Sciences* 27, 7 (2023), 597–600.

[12] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767* (2023).

[13] Pablo Villalobos et al. [n.d.]. *Will We Run Out of Data? Limits of LLM Scaling Based on Human-Generated Data.* Retrieved May 9, 2025 from https://epoch.ai/blog/will-we-run-out-of-data-limits-of-llm-scaling-based-on-human-generated-data

[14] Daniele Faraglia. 2025. Faker: A Python package that generates fake data. https://github.com/joke2k/faker. Version 37.3.0.

[15] Johann Frei and Frank Kramer. 2023. Annotated dataset creation through large language models for non-english medical NLP. *Journal of Biomedical Informatics* 145 (2023), 104478.

[16] Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610* (2022).

[17] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

[18] Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy F Chen, Zhengyuan Liu, and Diyi Yang. 2023. Coannotating: Uncertainty-guided work allocation between human and large language models for data annotation. *arXiv preprint arXiv:2310.15638* (2023).

[19] Xiao-Yang Liu, Guoxuan Wang, Hongyang Yang, and Daochen Zha. 2023. Fingpt: Democratizing internet-scale data for financial large language models. *arXiv preprint arXiv:2307.10485* (2023).

[20] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems* 36 (2023), 46534–46594.

[21] Microsoft. 2021. Azure Language Service. https://learn.microsoft.com/en-us/azure/ai-services/language-service/ Accessed: 2025-05-07.

[22] OpenAI. 2022. GPT-3.5: Generative Pre-trained Transformer 3.5. https://platform.openai.com/docs/models/gpt-3.5-turbo Accessed: 2025-05-07.

[23] OpenAI. 2024. GPT-4o: Optimized Variant of GPT-4. https://openai.com/research/gpt-4 Accessed: 2025-05-07.

[24] J OpenAI Achiam, S Adler, S Agarwal, L Ahmad, I Akkaya, FL Aleman, D Almeida, J Altenschmidt, S Altman, S Anadkat, et al. 2023. GPT-4 technical report. arXiv. *arXiv preprint arXiv:2303.08774* (2023).

[25] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[26] saak Uchakaev (Likid Geimfari: lk geimfari) and Mimesis Contributors. 2024. Mimesis: High-performance fake data generator for Python. https://github.com/lk-geimfari/mimesis. Version 18.0.0.

[27] Human Signal. [n.d.]. *Label Studio Documentation.* Retrieved May 9, 2025 from https://labelstud.io/guide

[28] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542* (2023).

[29] Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. *arXiv preprint arXiv:2402.13446* (2024).

[30] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).

[31] Uer Team. 2021. sbert-base-chinese-nli. https://huggingface.co/uer/sbert-base-chinese-nli Accessed: 2025-05-07.

[32] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[33] Sentence Transformers. 2020. LaBSE (Language-agnostic BERT Sentence Embedding). https://huggingface.co/sentence-transformers/LaBSE Accessed: 2025-05-07.

[34] Sentence Transformers. 2021. paraphrase-multilingual-mpnet-base-v2. https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2 Accessed: 2025-05-07.

[35] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560* (2022).

[36] Jianfei Wu, Xubin Wang, and Weijia Jia. 2025. Enhancing text annotation through rationale-driven collaborative few-shot prompting. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[37] Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594* (2024).

[38] Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. 2024. Self-distillation bridges distribution gap in language model fine-tuning. *arXiv preprint arXiv:2402.13669* (2024).