HILTS: Human-LLM Collaboration for Effective Data Labeling

Juliana Barbosa, Eduarda Alencar, Grace Fan, Aécio Santos, Juliana Freire

New York University

Abstract

The growing complexity and volume of data highlight the importance of learning-based classifiers across diverse tasks, from medical diagnosis to environmental monitoring. A common and impactful use case is data triage—efficiently identifying rare, relevant instances in large, imbalanced datasets. This is crucial for enabling domain experts to focus on what matters most. However, traditional supervised learning approaches often struggle with scalability due to the high cost and time required for manual labeling.

We introduce HILTS (Human-In-the-loop Learn To Sample), a framework designed to tackle these limitations. HILTS leverages Large Language Models (LLMs) for automated initial labeling and strategically incorporates human expertise through advanced active learning techniques. It selects diverse and representative samples for pseudo-labeling and identifies highly uncertain or likely incorrect LLM labels for targeted human review. This focused use of human effort maximizes the value of domain expertise while minimizing annotation overhead.

Our system reduces human labeling effort by up to 80% while outperforming few-shot foundation models such as GPT-4 by over 5% in F1-score in some scenarios—all at a significantly lower cost. HILTS also shows clear improvements over fully automated pseudo-labeling approaches and proves especially effective in handling class imbalance in real-world datasets. Its adaptability and efficiency make it a practical and scalable solution for high-stakes, domain-specific data triage tasks.

Email addresses: juliana.barbosa@nyu.edu (Juliana Barbosa), eduarda.alencar@nyu.edu (Eduarda Alencar), grace.fan@nyu.edu (Grace Fan), aecio.santos@nyu.edu (Aécio Santos), juliana.freire@nyu.edu (Juliana Freire)

Keywords: Human-in-the-loop, Large Language Models, Classification, Labeling, Data Triage

1. Introduction

Supervised machine learning methods, such as classification, have become indispensable tools for large-scale data analysis, particularly for *data triage* tasks where users need to identify and prioritize specific rare items within large data collections. This challenge spans numerous critical domains from monitoring online advertisements for illegal wildlife products [1], to detecting traces of criminal activity across web platforms [2].

In these scenarios, finding relevant information is akin to searching for needles in haystacks. Traditional search approaches, such as keyword-based queries, prove inadequate for such complex information needs. They lack the expressivity required to capture nuanced patterns and often return overwhelming numbers of irrelevant results due to lexical ambiguity and semantic variations.

Machine learning classifiers offer a more promising alternative by learning to distinguish relevant items based on complex feature patterns rather than simple keyword matching. Such classifiers can not only identify items that satisfy specific criteria but also rank them by confidence scores, enabling analysts to focus their limited attention on the most promising leads. This automated prioritization is particularly valuable when dealing with time-sensitive investigations or resource-constrained analysis scenarios.

However, the traditional supervised learning paradigm creates a significant bottleneck: *each new classification task requires curating a specialized training dataset*, demanding that domain experts manually examine a large number of examples to identify positive instances. This problem is compounded in domains in which relevant items are scarce, making up a small percentage of the data items. The labeling process is labor-intensive and time-consuming, severely constraining the scale, scope, and timeliness of critical analyses, often forcing researchers to examine only small data subsets or narrow time windows rather than conducting the comprehensive investigations needed to understand complex phenomena.

Data Triage for the Detection of Wildlife Trafficking. Wildlife trafficking is a global issue with dire environmental and health consequences, including significant biodiversity loss [3, 4, 5]. Despite increased efforts to

combat it, the rise of online marketplaces [6] has created new challenges for endangered species [7, 8, 9]. At the same time, traffickers leave digital footprints, which can be analyzed to gain insights into trafficking activities.

The main challenge lies in identifying relevant data among millions of online ads, as broad searches often yield irrelevant results [10]. For example, searching for "shark" on eBay may return ads for toys, clothing, and vacuum cleaners alongside items containing actual shark parts. Experts must carefully curate this data to ensure accuracy, but this manual process is time-consuming, limiting the scope of research on online wildlife trafficking [11, 12, 13, 14, 15, 16, 17, 18, 7, 19, 20]. The same challenge arises when trying to understand the overall landscape of animal products online. For instance, consider the challenge of curating training data for the following classifiers.

Example 1.1 (Animal Products)**.** An environmental scientist aims to build a classification model to identify any animal-derived product advertised online on an e-commerce platform. Understanding the overall landscape requires searching e-commerce platforms that contain millions of diverse product listings [21, 10]. Animal-related products, especially those associated with illicit trade, are exceedingly rare compared to other irrelevant animal-related products (e.g., plush toys, postcards, photos, etc).

Example 1.2 (Small Leather Products)**.** To focus specifically on identifying the illegal trade of small leather goods made from protected species—such as belts, wallets, and bags made from alligator or snake skin—it is necessary to develop an even more specialized classifier. These items represent a tiny fraction of not only all e-commerce ads but also general leather goods. The manual effort required to build labeled data that distinguishes these rare, specific items from legitimate leather is highly time-consuming and inevitably results in imbalanced training dataset.

For each of these tasks, experts would ideally label a large, representative dataset to train a robust classifier. However, the stark differences between these tasks often mean that classifiers are not reusable, greatly limiting the experts' ability to explore multiple research questions simultaneously.

Limitations of Few-Shot Classification using LLMs. Foundation models, particularly Large Language Models (LLMs), offer an alternative solution due to their vast knowledge and natural language understanding capabilities

[22, 23, 24]. They can perform few-shot classification, significantly reducing the need for human input. However, classifying millions of ads directly with powerful LLMs like GPT-4 is prohibitively expensive (e.g., it costs over $17,000 to classify 800,000 ads). This cost, combined with the environmental impact of LLMs [25] and the high percentage of irrelevant data, makes direct zero-shot classification impractical for most researchers and under-resourced institutions [26]. While open-source models (such as Llama3, LLaVA) offer lower costs, their accuracy is often significantly lower [10].

Leveraging LLMs for Cost-Effective Labeling. To support cost-effective classification of data for diverse research questions, we previously introduced LTS (Learn to Sample) [10]. LTS addresses the cost challenge by using LLMs to annotate a subset of the collected data using few-shot in-context learning. These LLM-generated pseudo-labels are then used to train smaller, specialized classifiers that can be applied at scale, drastically reducing the total number of expensive LLM inferences needed.

A core challenge for LTS is developing effective sampling strategies for highly imbalanced datasets where the target class represents a tiny fraction of available data. Random sampling proves inadequate in this setting, as it fails to capture sufficient positive examples for meaningful model training while simultaneously over-representing the majority class, leading to classifiers that exhibit poor recall on the rare but critical instances that analysts actually seek to identify. LTS tackles this problem by learning to select samples: it clusters the data to ensure diversity and employs a multi-armed bandit strategy to balance exploration (finding new types of ads) and exploitation (sampling from clusters likely to contain relevant ads), all within an iterative active learning framework. This strategy allows LTS to build and refine models to classify and select relevant ads efficiently. Our previous work demonstrates that models trained with LTS-labeled data achieve high accuracy and have significantly reduced costs compared to direct LLM classification.

However, a key limitation of LTS is its reliance on a manually curated validation set, which guides the sample selection process. Constructing such a set is non-trivial: the user must identify a subset of data that is both diverse and representative of the overall distribution—despite limited prior knowledge—and manually label each item. If the validation set is biased or lacks sufficient coverage of relevant subtypes, it can mislead the sampling strategy, resulting in poor model performance and inefficient use of labeling

effort (i.e., the user may waste time labeling ads that do not significantly improve performance).

While LTS significantly reduces the cost and effort compared to manual labeling or direct LLM inference on full datasets, it relies solely on LLM-generated pseudo-labels. Although LLMs are effective at providing initial labels, they are not foolproof, especially efficient LLMs that have only a few billion parameters. When LLMs make mistakes, especially systematic ones, those errors can be propagated through the iterative learning process. Incorrectly labeled data can degrade the training signal provided to the classifier, leading to poor sampling decisions and, ultimately, an ineffective classifier. By incorporating human expertise into the labeling process, we can further enhance label quality and, consequently, model performance (as we demonstrate in Section 5).

Contributions. In this paper, we introduce the *HILTS framework*. HILTS framework generalizes the LTS algorithm, introducing the human in the loop. It is designed to minimize the amount of human effort required to build training data for classifiers, while enhancing the quality of the labeled data produced. As opposed to LTS, which requires a manually curated validation dataset, HILTS offers flexibility regarding validation data: it allows users to provide an existing validation set, if available, but it can also dynamically generate one automatically. Users provide a detailed task description and parameters that are used to create LLM-generated pseudo-labels, thus integrating user domain knowledge directly into the label generation. To reduce the number of pseudo-labels reviewed by humans, HILTS carefully sub-samples these labels and presents them to users who can review and correct them if necessary, ensuring that domain-specific nuances and critical knowledge are accurately captured. This human feedback loop is crucial for refining training data and building more accurate and reliable classifiers.

To facilitate human in the loop, the framework supports different sampling strategies designed to provide a smaller and highly-informative sample for user review. These include: (1) random selection, where a user-specified sample size is randomly drawn; (2) an embedding-based approach that identifies potential label disagreements or areas of interest based on the similarity of the samples; and (3) an uncertainty-based approach, which leverages the trained model to predict labels and prioritize samples where the model's confidence is lowest. More comprehensive details about these three sampling approaches will be described in Section 3.4.

To streamline the incorporation of human feedback and support the use cases in Examples 1.1 and 1.2, we implement the *HILTS framework* in the *HILTS system*, which allows users to build training data seamlessly through an easy-to-use interactive user interface. This user interface (UI) provides users with detailed control over input settings, enabling them to easily define tasks, configure all core framework components, upload their data, and leverage powerful tools for data analysis, exploration, and labeling. These capabilities, which include embedding-based semantic search and visualization of records in the input dataset, facilitate comprehensive data exploration of the input dataset. In addition, it guides users through the HILTS data labeling process, streamlining the generation and review of labeled data.

To evaluate HILTS, we conduct an experimental evaluation using two real research questions (described in Examples 1.1 and 1.2) with varying data collection requirements, each with different collection sizes (200k and 700k ads) and complexity levels (from specific to generic questions). Our experiments demonstrate that HILTS provides a scalable and cost-effective solution for generating high-quality training data that significantly improves the quality of models trained by pseudo-labeled data generated by an open-source LLM. HILTS achieves substantial performance gains, outperforming state-of-the-art LTS approach by up to 20% in F-1 score and surpassing few-shot classification with commercial LLMs by over 5% at a fraction of the cost.

Our main contributions can be summarized as follows:

- We propose HILTS, a novel human-in-the-loop framework that generalizes the LTS approach by integrating user domain knowledge into the iterative LLM-based pseudo-labeling process, without requiring initial human-validated data. This framework offers validation set generation and various sampling strategies to efficiently prioritize samples for human correction. The main goal is to minimize the amount of work required of users while increasing pseudo-labeling quality.

- We develop an interactive user interface that implements the *HILTS framework*, allowing users to easily define tasks and configure components of the core framework. The interface also provides additional powerful features for data exploration and analysis.

- We perform an experimental evaluation, showing that HILTS provides a scalable and cost-effective solution for generating high-quality training

data and outperforms state-of-the-art methods and few-shot classification using commercial LLMs.

2. Related Work

Our work on the *HILTS framework* draws inspiration from and extends several key areas of machine learning and data management, particularly those addressing the challenges of data labeling, active learning, and the application of Large Language Models (LLMs) in data-scarce and imbalanced scenarios.

Labeling Data to Create Classifiers. Building effective learning-based models, especially for classification tasks, remains fundamentally constrained by the availability of high-quality training data. The database community has long recognized this bottleneck, proposing various approaches to reduce human labeling effort [27, 28, 29, 30, 31]. Methods like Snorkel [31] introduced data programming, in which users write heuristic labeling functions, while Snuba [30] aimed to automate heuristic generation from small labeled sets. Others, such as Inspector Gadget [29] and Goggles [27], have focused on weakly labeling image datasets. For a comprehensive overview, see the survey by Whang et al. [28].

The *HILTS framework* builds upon and significantly extends the Learning-to-Sample (LTS) approach [10]. Like LTS, HILTS leverages LLMs for scalable data labeling, utilizing their broad knowledge as general-purpose classifiers without requiring users to write explicit labeling functions or provide initial human-labeled data. Instead, users define the classification task via a prompt. However, while LTS demonstrated a cost-effective way to use expensive LLMs sparingly by training smaller, cheaper models, HILTS specifically addresses the inherent limitations of relying solely on LLM-generated pseudo-labels, such as the introduction of significant noise that can degrade downstream classifier performance. HILTS tackles this by introducing a crucial human-in-the-loop component and sophisticated sampling strategies, ensuring higher labeling accuracy while still maintaining LLMs as the primary labeling agents.

Online Data Triage and Wildlife Trafficking Detection. Our work directly applies to data triage problems across numerous domains, where users seek to identify and prioritize rare items within massive online data collections. A prime example, and the driving application for HILTS, is the detection

of online wildlife trafficking. This domain highlights the severe challenges in finding information that meets particular criteria.

Existing research in wildlife trafficking detection, as highlighted by Keskin et al. [32], consistently points to data scarcity and bias as critical challenges. Studies like Cardoso et al. [33] and Xu [34] demonstrate the effectiveness of models focused on specific species (e.g., pangolins) or products (e.g., ivory), while Kulkarni and Di Minin [35] emphasize the difficulty of obtaining high-quality training data for broader applications. Extensive efforts have been made in wildlife image detection [34, 33, 35, 36, 37, 38, 39] and text classification for specific wildlife contexts [40, 41]. For a broader perspective, see Tuia et al. [42]. These studies consistently underscore the need for large training data volumes and the generalization limitations of current models across different species or products. HILTS directly addresses these challenges by providing a scalable and cost-effective solution for generating high-quality training data, capable of identifying rare instances and improving the quality of smaller, open-source models, even for highly imbalanced datasets where domain expertise is crucial.

Active Learning for Imbalanced Data. Dealing with class imbalance—where the number of examples in different classes varies significantly—is still a challenge in machine learning [43, 44, 45, 46, 47, 48]. Traditional strategies typically involve selecting subsets from already labeled instances through methods like random sampling, over-sampling, under-sampling [49, 45], or applying clustering algorithms [46]. However, in our context of rare product detection, where labels are initially unknown, these methods cannot be directly applied.

When unlabeled data is abundant and labels are expensive, *active learning* (AL) methods offer a solution by iteratively selecting the most informative data points for an oracle (human or model) to label [50, 51]. Common AL strategies include *uncertainty sampling* [52, 53, 54], which prioritizes instances the oracle is most uncertain about, and *diversity sampling* [55], which aims for a representative sample covering a wide range of features. HILTS combines and extends principles from both class imbalance handling and active learning to effectively discover rare instances in large, unlabeled datasets. Prior work [56, 57, 58] has used active learning techniques to uncover rare instances, such as Aggarwal et al. [58], who use active learning to iteratively refine the classifier to promote minority classes. Similar to this work, HILTS leverages model predictions but distinguishes itself by clustering unlabeled

instances for initial diversity and employing a multi-armed bandit algorithm to select instances from different clusters to promote the discovery of rare positive examples. While using similarity measures and core-sets for diversity is a direction for future work, our current approach has proven effective in deriving high-quality classifiers at a low cost, as demonstrated in Section 5.

LLM-based Active Learning. Recent advancements in LLMs have introduced novel approaches to reduce the cost and effort of active learning, with methods categorized into LLM-based selection, generation, annotation, and hybrid strategies [59]. Methods such as ActiveLLM [60] leverage LLMs to perform unsupervised instance selection in few-shot settings, reducing reliance on traditional acquisition functions. Similarly, [61] explores hybrid active learning for neural machine translation, demonstrating how combining automatic labeling with selective human intervention can improve efficiency. Papers such as FreeAL [62] framework use a fully automated approach, entirely eliminating the human annotation bottleneck, by establishing an internal feedback loop where an LLM acts as an active annotator and a smaller language model (SLM) serves as a student filter, identifying high-confidence pseudo-labels and feeding them back to the LLM as high-quality examples. HILTS falls into the hybrid systems category, which includes frameworks such as NoiseAL [63] that also highlight the benefits of limited human verification but fall short in identifying the challenges of active learning on imbalanced data. Our proposed HILTS framework advances this line of work by explicitly targeting the challenges of imbalanced datasets and rare positive discovery, where naive reliance on pseudo-labels risks propagating systematic errors. Unlike selection-focused methods [60] or domain-specific hybrid pipelines [61], HILTS integrates multi-armed bandit sampling and uncertainty-driven human verification to ensure label quality, positioning it as a scalable and robust human–LLM collaborative active learning approach for real-world data triage tasks.

Using LLMs as Evaluators and Annotators. The increasing sophistication of LLMs has positioned them as powerful alternatives to human evaluators and annotators across a variety of tasks, from evaluating natural language generation (NLG) systems [64, 65, 66, 67] to performing general classification for tasks like column-type annotation and schema matching [68, 69]. In large-scale data triage applications, such as identifying wildlife-related ads, the sheer volume of data makes human-only labeling prohibitively expensive. HILTS strategically leverages LLMs to generate initial pseudo-labels, forming

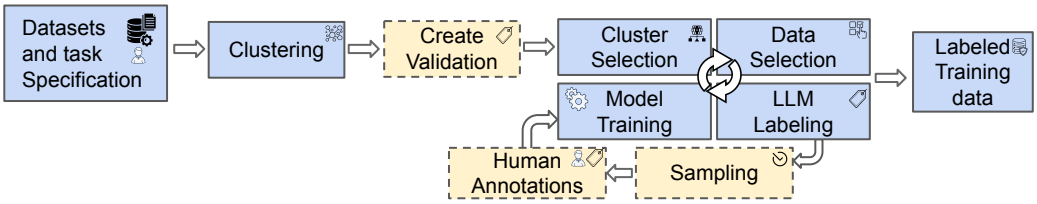


Figure 1: HILTS pipeline. The user first provides a dataset and specifies their task configurations. The process starts with clustering. A validation set is generated and labeled by an LLM if the user provides no validation set. Next, the active learning process starts, in which the data is sampled and labeled by an LLM. After the LLM finishes labeling, a small sample is selected to be reviewed by the user. The yellow dashed boxes are steps in the pipeline where different choices can be made by the user.

the foundation of its training data creation process.

Human-LLM collaboration. Large Language models (LLMs) have been widely used for automated data annotation [70]. However, relying solely on LLM-generated pseudo labels can introduce biases and errors, particularly in specialized domains like wildlife trafficking detection. Recent studies have explored human-LLM collaboration to improve annotation accuracy [71, 72, 73]. Pangakis & Wolken [74] highlight the importance of keeping humans in the loop for automated annotation, showing that user oversight significantly improves label quality and model performance. HILTS builds on this insight by incorporating human validation, ensuring that LLM-generated pseudo labels are reviewed and corrected before further refining the classifier. This hybrid approach allows for cost-effective scaling while mitigating the inherent limitations of fully automated LLM labeling.

3. HILTS Framework: Hybrid Human-LLM Data Labeling

3.1. Framework Overview

As highlighted in Section 1, finding rare items in large data collections, such as wildlife products in e-commerce advertisements datasets, is incredibly expensive and time-consuming. This is partially due to the need to obtain high-quality labeled training data to train classification models. Manual labeling is time-consuming, and while foundation models (such as LLMs) can perform zero-shot classification, doing so directly on massive datasets is computationally expensive and may lack the required precision for nuanced tasks compared to specialized models.

Algorithm 1 HILTS Framework

Initialize Budget B, Size of sample s, *metric_baseline*, and sampling approach A
$M_{current} \leftarrow$ Base model
$C \leftarrow$ clustering(D, k), where D is the data collection, k is the number of clusters
Validation Set $V \leftarrow get\_or\_create\_validation\_set(C)$
Training Data $T \leftarrow \varnothing$
for $i = 1$ to B **do**
    $current\_sample \leftarrow \varnothing$
    if $i == 1$ **then**
        for each cluster c_i in C **do**
            $t_i \leftarrow$ Select a sample from c_i of size $\lfloor s/k \rfloor$
            $current\_sample \leftarrow current\_sample \cup t_i$
        end for
    else
        $c \leftarrow$ thompson_select_cluster()
        if $M_{trained}$ **then**
            $c_{pos\_indices}, c_{neg\_indices} =$ predict_labels(M, c)
            $t_{pos} \leftarrow$ Sample from c at $c_{pos\_indices}$ of size $\lfloor p \cdot s \rfloor$
            $t_{neg} \leftarrow$ Sample from c at $c_{neg\_indices}$ of size $\lfloor (1-p) \cdot s \rfloor$
            $current\_sample \leftarrow t_{pos} \cup t_{neg}$
        else
            $current\_sample \leftarrow$ Sample from c of size s
        end if
    end if
    $current\_labeled\_sample \leftarrow$ get_labels$(current_sample, A)$
    $t_{pos}, t_{neg} \leftarrow$ Split$(current_labeled_sample)$
    thompson_update_reward(t_{pos}, t_{neg}, c)
    $M_{trained} \leftarrow$ fine_tune$(M_{current}, current_sample_labeled)$
    $metric \leftarrow$ evaluate$(M_{trained}, V)$
    if $metric > metric\_baseline$ **then**
        $M_{current} \leftarrow M_{trained}$
        $metric\_baseline \leftarrow metric$
    end if
end for

The *HILTS framework* builds upon the Learn-to-Sample (LTS) approach [10], a cost-effective strategy designed to generate labeled data to train specialized classifiers that identify specific data instances in large, imbalanced datasets. The core idea behind LTS is to strategically leverage powerful LLMs not for large-scale direct classification, but as pseudo-labelers on a carefully selected subset of the data. This pseudo-labeled subset is then used to train smaller, more efficient classification models tailored to a specific task that can be deployed at a fraction of the cost of repeatedly querying LLMs for every data point.

Another characteristic of *HILTS framework* is that it incorporates human-in-the-loop supervision into the labeling pipeline, thus improving labeling quality and allowing domain experts to review LLM-generated pseudo-labels. Crucially, *HILTS framework* integrates an additional step: sampling a smaller set of these LLM-labeled instances for human revision. This comprehensive approach aims to learn a high-performing classification model while minimizing both human effort and computational costs.

The overall pipeline is outlined in Algorithm 1 and is visually summarized in Figure 1. The framework starts by setting the datasets, task specification and key parameters: a predefined total number of iterations B (determined based on an overall labeling budget), a chosen sample size s for each iteration, a metric baseline (precision, recall, F1 or accuracy), the sampling approach A if the user is reviewing the LLM labels and a pre-trained base model (i.e. bert-base-uncased) that is used as the first model to be fine-tuned (lines 1–2).

The entire data collection D is initially *clustered* into k groups (line 3). This step is performed once and aims to promote sampling diversity and enable exploration across different semantic regions of the data space.

A crucial component of the process is the validation set (V), which allows the evaluation of the model's performance throughout the AL iterations. The function `get_or_create_validation_set(C)` will get the validation set provided by the user with human-annotated labels, or create a validation set from a random sample, selected from each of the clusters, without replacement.

The main component of the algorithm is the active learning process that runs for B iterations (lines 6–33). The first iteration begins with samples drawn uniformly across clusters to ensure broad initial coverage (lines 8–12). In subsequent iterations, cluster selection is governed by a multi-armed bandit (MAB) strategy, specifically, Thompson Sampling with the function `thompson_select_cluster()` (line 14), which effectively balances

exploration (sampling from less-certain or underexplored clusters) and exploitation (focusing on clusters known to contain relevant data).

If a classifier model has been trained, we use this model to predict the labels of samples within a selected cluster with the function `predict_labels(`M`,`c`)` (line 16). A proportion p of the sampled items is then drawn from those predicted as positive (exploitation), with the remaining $(1 - p)$ drawn from likely negatives (exploration within the chosen cluster) (lines 17–19). This biased sampling allows the framework to increase the chances of selecting rare (positive) samples while still discovering novel data patterns.

In the `get_labels(`*current_sample*`,` A`)` function, each sampled batch undergoes *pseudo labeling* using an LLM and few-shot prompting. The prompt includes a natural language description of the task criteria and a few examples to guide the LLM in generating a pseudo-label for each sample. In cases where user feedback is provided, the function also samples from the pseudo-labeled data based on three different sampling strategies A: random, nn-voting, or uncertainty-based (more details on these strategies will be discussed in Section 3.4). These samples are shown to the user, who can then correct or validate the labels (line 24).

Since the LLM can also label the validation set, the labeled and reviewed data are used to update the reward signal for the multi-armed bandit (MAB) policy, which guides cluster selection. The MAB reward is now directly based on the abundance of rare positive instances found within the selected sample. The function `thompson_update_reward(`t_{pos}`,` t_{neg}`,` c`)` receives the number of positive samples t_{pos} and negative samples t_{neg} selected from cluster c (lines 25–26). This reward strategy means the MAB learns to identify and sample from clusters that are rich in the types of data desired for training (e.g., clusters containing a higher density of positive examples or a good mix), rather than relying solely on the indirect signal of incremental model performance improvement, since the model is now being evaluated on a validation set that are not generated from "golden" label. This ensures that the active learning process effectively discovers and includes sufficient quantities of the target rare class examples, providing a stronger foundation for subsequent human review and final model training.

Finally, the labeled batch is used to fine-tune the current classifier (line 27). The newly trained model is evaluated on the validation set (line 28), and if its performance improves over the established baseline, the model is updated and carried forward to the next iteration (lines 29–32).

By structuring sampling and training as an iterative loop guided by real-

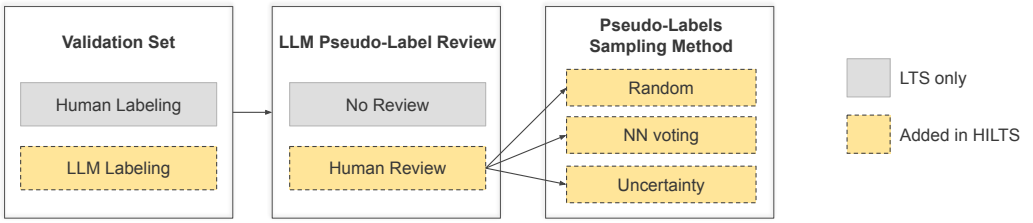


Figure 2: The HILTS framework generalizes LTS by introducing a human-in-the-loop at key parts of the data labeling process, and providing flexible options for supporting which samples are shown to the human labelers. Gray boxes represent the original LTS implementation, while the rest are introduced in HILTS.

time model feedback and active selection, HILTS generalizes the LTS approach into a modular and extensible framework. While the core principles of the Learn to Sample (LTS) algorithm remain foundational, HILTS offers flexibility in how certain key components are handled, enabling different instantiations based on the availability of humans to label data and application needs. As shown in Figure 2, this design accommodates various sources of model validation set (e.g. Human labeling or LLM labeling), an option to review LLM labels (e.g. No review or Human review), and different pseudo-labels sampling methods (e.g. Random, NN voting, and Uncertainty). The three design components will be explained in depth in Sections 3.2, 3.3, 3.4.

3.2. Validation Set

Human-Labeled Validation Data. Figure 2 shows all the possible instantiations of *HILTS framework*. One such instantiation, reflecting the approach of the original LTS framework, evaluates model performance during active learning iterations against a small set of manually labeled "validation gold data". This method uses a reliable human-labeled benchmark to track model improvement and calculate the reward signal for cluster selection.

LLM-Labeled Validation Data. The approach primarily adopted within the *HILTS framework* for its automated efficiency adapts to how the validation data is obtained. Instead of relying on a potentially costly, manually curated "gold" set for validation, a validation dataset is automatically generated. This comprehensive sample, spanning a fixed number of ads from each of the clusters, is then labeled using an LLM, similar to how training samples are pseudo-labeled. This LLM-labeled validation set, created once at the beginning, serves as a benchmark to evaluate the improvement of the incrementally

trained model in each active learning iteration. This approach ensures that the validation set is representative of the diversity across all clusters and can be generated without requiring additional human labeling effort specifically for validation.

3.3. LLM Pseudo-Labels Review

Even though *HILTS framework* is created to address the need for humans to curate possible LLMs' errors in the labeling process, it also allows for a fully automated version where the training data is created without any human intervention. As already demonstrated with the LTS algorithm, the ability of Large Language Models (LLMs) to perform classification tasks in a zero-shot or few-shot manner presents a compelling opportunity to automate the labor-intensive process of data labeling.

Enhancing Data Curation with Human-in-the-Loop. However, relying solely on this approach to generate training data faces some limitations, such as the introduction of significant noise, which can limit the performance of the downstream classifier. In such cases, *HILTS framework* integrates the human as a critical quality control layer, directly engaging them in validating and correcting LLM-generated labels. In those cases, the framework includes three primary sampling methods to select which data will be provided to the user for manual revision. The sampling is performed over the data selected on that specific cycle, after being labeled by the LLM.

3.4. Pseudo-Labels Sampling Methods for Human Review

Rather than presenting users with all the pseudo-labeled data for verification (e.g., 200 data items labeled by an LLM in our experiments, as shown in Section 5), HILTS supports different strategies to select the most informative samples that users could assess to improve the labeled data.

Random Sampling Selection. In this simple approach, a batch of samples is randomly selected from items that have already been pseudo-labeled by the large language model (LLM). The batch size is a parameter defined by the user. The goal is to provide the user with an unbiased sample of the original data distribution.

Nearest Neighbor Voting Sample Selection. We also introduce a verification strategy that leverages user-reviewed data and embedding-based similarity. For each LLM-labeled data point, we retrieve a list of similar examples using precomputed embeddings and vector search. The similarity scores are

converted to distances, and the resulting sorted distances are analyzed using the Kneedle algorithm [75] to determine an appropriate cutoff K. This adaptive selection ensures that only a meaningful set of neighbors—those most informative yet diverse—are used. A majority vote over the labels of the top-K neighbors is then compared to the LLM's prediction. If the votes contradict the LLM label, the data point is flagged for user revision. If the total number of flagged items is insufficient to meet a user-defined sample size, additional examples are chosen via random sampling.

Let $\mathcal{D}_{\text{LLM}} = \{(x_i, \hat{y}_i)\}_{i=1}^{n}$ be the set of data points labeled by the LLM, and let $\mathcal{D}_{\text{user}} = \{(x_j, y_j)\}_{j=1}^{m}$ be the set of data points labeled and confirmed by the user in the previous iterations. Each data point x has a precomputed embedding $\phi(x) \in \mathbb{R}^d$.

1. For each $x_i \in \mathcal{D}_{\text{LLM}}$, retrieve its nearest neighbors from $\mathcal{D}_{\text{user}}$ using vector search:

$$\mathcal{N}(x_i) = \{(x_j, y_j) \in \mathcal{D}_{\text{user}} \mid x_j \text{ is among top similar points to } x_i\}$$

2. Compute distances using cosine similarity:

$$d(x_i, x_j) = 1 - \cos(\phi(x_i), \phi(x_j))$$

3. Sort the distances in increasing order:

$$d_1 \leq d_2 \leq \cdots \leq d_w$$

4. Use the Kneedle algorithm to determine the cutoff K such that the top-K neighbors capture the "knee" in the distance curve:

$$K = \texttt{Kneedle}(d_1, \ldots, d_w)$$

5. Let the top-K neighbors be:

$$\mathcal{N}_K(x_i) = \{(x_j, y_j) \in \mathcal{N}(x_i) \mid j \leq K\}$$

6. Perform a majority vote over their labels:

$$\tilde{y}_i = \texttt{MajorityVote}(\{y_j \mid (x_j, y_j) \in \mathcal{N}_K(x_i)\})$$

7. If $\tilde{y}_i \neq \hat{y}_i$, then x_i is flagged for user revision.

8. If fewer than the desired number of flagged points are selected, fill the remaining quota using uniform random sampling from $\mathcal{D}_{\text{LLM}}$.

Random sampling is performed in the first iteration, since no data is labeled at the time. For this approach, the idea is to increase the chances of correcting more samples by comparing them to previously user-verified data.

Uncertainty-Based Sample Selection. For the uncertainty-based sampling approach, the model fine-tuned in the previous iteration is used to infer a label for the set labeled by the LLM. The model returns the logits and a softmax that are applied to retrieve the probability of each class. The most uncertain data is ranked, and the top K (which is also a number pre-selected by the user) is returned for verification.

Let M_t be the classifier model fine-tuned at iteration t, and let $\mathcal{D}_{\text{LLM}} = \{x_i\}_{i=1}^{n}$ be the set of unlabeled samples with LLM-predicted labels.

1. For each $x_i \in \mathcal{D}_{\text{LLM}}$, compute the class probability distribution using the model's softmax output:

$$\mathbf{p}_i = \texttt{softmax}(M_t(x_i)) \in [0,1]^C$$

   where C is the number of classes ($C = 2$ for binary classification).

2. Define the uncertainty score for x_i as:

$$u(x_i) = |\max(\mathbf{p}_i) - 0.5|$$

   This formulation reflects how close the model is to being undecided (maximum uncertainty at 0.5) and is lowest when the model is most uncertain.

3. Rank the data points by increasing uncertainty:

$$\mathcal{D}_{\text{sorted}} = \texttt{sort}(\mathcal{D}_{\text{LLM}},\ \text{by } u(x_i))$$

   This ensures that the most uncertain (i.e., lowest $(u(x_i))$ samples come first and are selected for user verification.

4. Select the top-K most uncertain samples for user review:

$$\mathcal{D}_{\text{uncertain}} = \{x_i \in \mathcal{D}_{\text{sorted}} \mid i \leq K\}$$

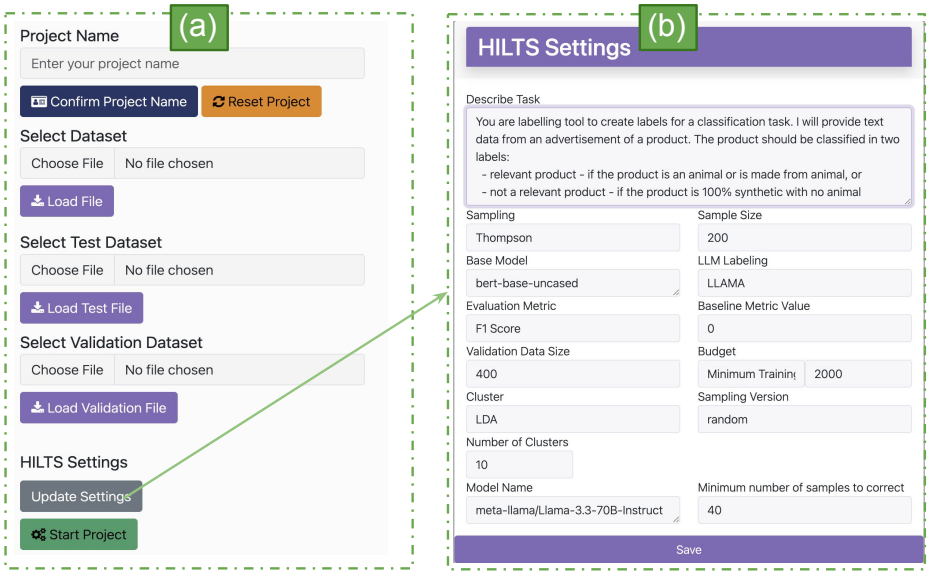


Figure 3: Initial HILTS Settings and Task Description Interface. (a) illustrates the user's ability to upload datasets and (b) configure core parameters such as budget, sample size, LLM model, base model for fine-tuning, evaluation metric, and clustering algorithm.

This approach prioritizes samples that the model is least confident about, thus refining the decision boundary and improving the performance on ambiguous cases.

4. The HILTS System

We develop the HILTS system that implements the proposed framework described in Section 3.1. In this system, we aim to support domain experts in building machine learning classifiers from their unlabeled dataset, providing an interactive user interface that allows users to: (1) describe a classification task and set HILTS parameters, (2) review pseudo-labels, (3) track model performance, and (4) explore data.

Classification task description and HILTS parameters. As illustrated in Figure 3(a), the user first uploads all available datasets, including the primary unlabeled data and optional validation and test sets. After the upload, a configuration interface for HILTS settings appears (shown in Figure 3(b)), offering a range of customization options. The most important setting is the task description, which serves as a prompt for the LLM, defining the specific

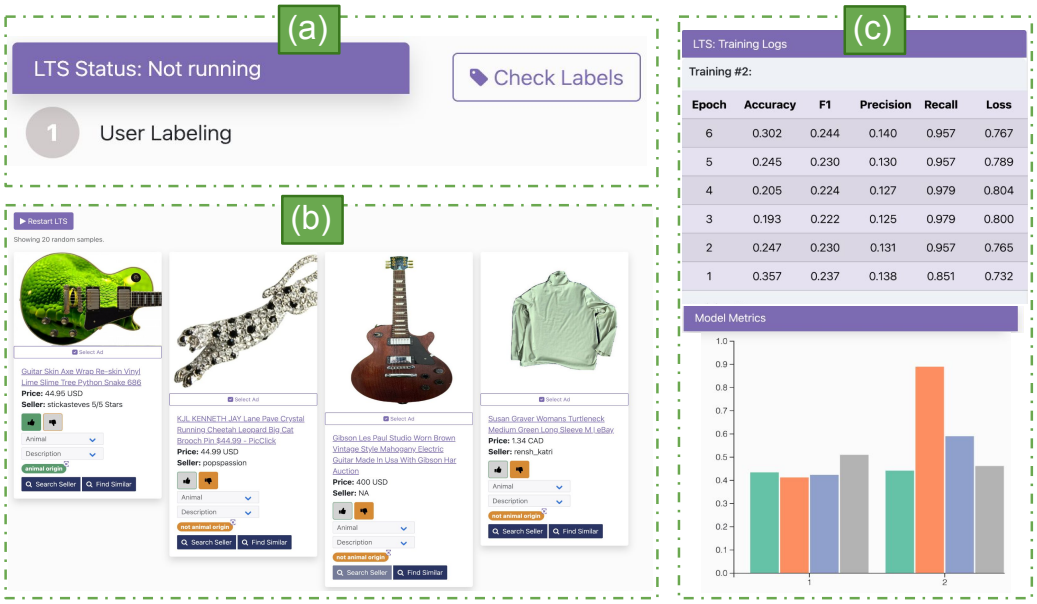


Figure 4: HILTS interface for (a) flag for users to review and correct LLM-generated pseudo-labels. (b) provides a list of cards representing each product with the LLM label already selected, and (b) provides the model results during the training phase.

classification goal for the data. Crucially, users can choose the "*Sample Size*" to manually review during the human-in-the-loop phase. This parameter provides direct control over the level of human effort and intervention, allowing for a balance between cost-effectiveness and desired accuracy. For tasks requiring higher precision, a larger number of samples can be selected for expert review. Users can also choose the "*Sampling Version*" – the method used to choose pseudo-labeled data for human review (Section 3.4). Finally, users can configure the underlying algorithm details, including the Large Language Model (LLM) used for automated labeling and the base machine learning model that will be fine-tuned during the active-learning phase. This level of control allows users to balance computational resources, API costs, and the desired level of labeling accuracy for their specific domain, ensuring the most appropriate components for the task at hand.

Pseudo-labels review. If the user configures the system to include human review (by setting a minimum number of samples to review and the algorithm for sampling), the interface will display a "*User Labeling*" label after each LLM labeling iteration, as illustrated in Figure 4 (a). Upon clicking

the "*Check Labels*" button, the data selected by one of the three sampling approaches (described in Section 3.4) will be presented on individual cards as seen in Figure 4(b). Each card showcases the item's text, its associated image (if available), and any other relevant information extracted from the item's metadata. The initial label displayed on each card (a green thumbs up for positive and an orange thumbs down for negative) is the pseudo-label provided by the LLM. Users can then modify this label if they disagree with the LLM's prediction. Once all labels have been reviewed, the user can click "*Restart*" to re-initiate the active learning process with the corrected data incorporated.

Model performance tracking. Upon restarting the active learning process, the model initiates its training phase, using a combined dataset of user-corrected data and LLM-labeled samples. Training results for each epoch are displayed on-screen. Subsequently, evaluation results are presented in a final metrics bar chart, drawing from the test set—if provided—or using the performance of the validation set, as illustrated in Figure 4(c).

Data exploration. In scenarios where users wish to generate additional labeled data outside the framework's active learning loop, the HILTS system offers dedicated data exploration components. These components display data in the same intuitive card format used for reviewing pseudo-labels, allowing users to add labels directly. Users can perform a random search to get a general idea about the data, or a keyword search to find more specific data via string matching. For example, as illustrated in Figure 5(a), the user can type the keyword "gator", and the system will return a list of data containing that term. If the user wants to find similar items to any of these results, they can click the "*Find Similar*" button. This action triggers a search for comparable items based on their smallest embedding vector distances, thereby facilitating a deeper exploration of the dataset, as demonstrated in Figure 5(b).

Implementation Details. HILTS is implemented in Python 3.11.2 and the web-based interface is implemented with svelte.js [76] (code is publicly available[1]). The embeddings generated for each data point are derived from CLIP embeddings (`clip-vit-base-patch32`) [77], which are numerical representations of images and text that exist within the same semantic space. The CLIP method generates distinct embeddings for both the image and text

[1] `https://github.com/VIDA-NYU/mmdx-wildlife/tree/wildtracker`

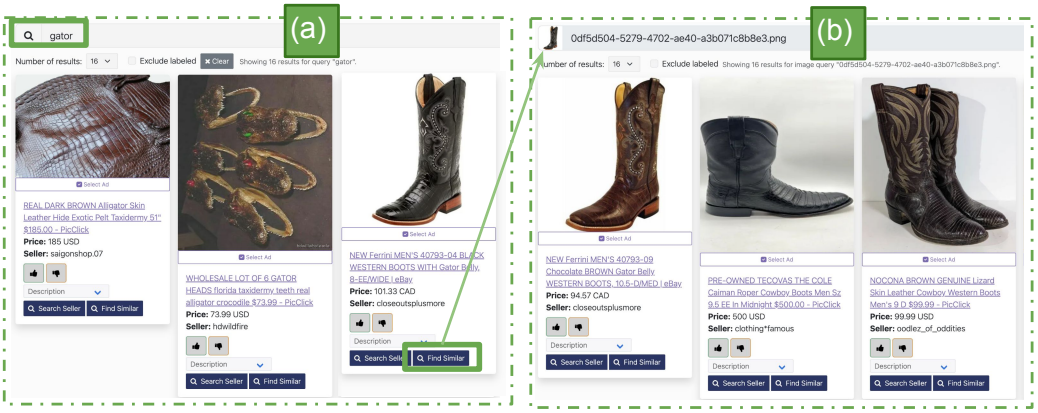


Figure 5: The HILTS System's Data Exploration Interface. (a) illustrates the Keyword Search functionality, allowing users to find specific data points via string matching (e.g., 'gator'). (b) demonstrates the 'Find Similar' feature, which leverages embedding distances to discover comparable items.

components of an advertisement, if available, and we compute their mean to form a single, comprehensive embedding. If the image is not available, only the text embedding is used. While CLIP embeddings have been widely adopted to embed images and text [78], an alternative embedding method can easily be substituted. The embeddings are stored and queried using LanceDB [79]. To make it easy to deploy, HILTS is entirely containerized using Docker, and the dataset and images (if available) can be stored locally or on an S3-like storage. HILTS applies LTS using its publicly available code.[2]

5. Experimental Evaluation

As introduced in Section 1, the online trade of wildlife products constitutes a significant global issue, making the data triage task to accurately identify these criminal activities particularly challenging. To evaluate our implementation of HILTS, we leverage a real-world dataset crawled from e-commerce platforms [21]. Our experiments are designed to explore two distinct research questions, reflecting the complexity and varying data characteristics highlighted in Example 1.1 (Animal Product Identification) and Example 1.2 (Leather Products) from Section 1.

[2] `https://github.com/VIDA-NYU/LTS`

	Validation Set	Label Review	Sampling
LTS	Manual	None	None
HILTS-Auto	LLM	None	None
HILTS-Random	LLM	Human	Random
HILTS-NN-Voting	LLM	Human	NN-Voting
HILTS-Uncertainty	LLM	Human	Uncertainty

Table 1: Summary of experimental configurations under different instantiations of the HILTS framework. The table outlines whether the validation set is manually labeled or generated by the LLM, whether pseudo-labels are reviewed by the user, and the sampling method used for selecting data for user review when applicable.

Data Collection. We utilize the open-source ACHE crawler to conduct a scoped crawl [80], in which given a list of seed webpages, the crawler recursively follows all hyperlinks that remain within a specified domain.

Animal products. The animal product ads collection consists of 699,907 valid listings, collected over one month from 33 different e-commerce platforms. The crawl targets ads related to mammals, birds, sharks and reptiles, and includes items associated with 263 endangered species listed under CITES [81].

Leather products. To compile seeds for leather-related products, we use the names of 48 animals identified in a dataset of seized wildlife items and their intended uses [82]. Using these terms, we retrieve ads from eBay [83], resulting in a collection of 152,495 valid listings.

By evaluating HILTS across these two distinct real-world scenarios, we aim to demonstrate its effectiveness in handling varying dataset sizes, class imbalances, and semantic complexities, showcasing its adaptability and efficiency in generating high-quality labeled data for challenging classification tasks.

5.1. Experimental setup

We compare various algorithmic configurations of the HILTS framework, as outlined in Table 1. The results are presented in Tables 2 and 3, focusing on metrics such as precision, recall, accuracy, and F1-score.

Our primary baseline for all new HILTS configurations is the original LTS approach, as it has already demonstrated superior performance compared to various sampling methods and other active learning algorithms [10].

These instantiations cover a spectrum of approaches, from fully automated to human-in-the-loop strategies with different sampling methods:

	Accuracy	Precision	Recall	F1	# Correct
HILTS-Random	0.946	0.803	0.970	0.879	28/400
HILTS-NN-Voting	0.938	0.850	0.842	0.846	122/400
HILTS-Uncertainty	0.926	0.762	0.921	0.834	63/400
HILTS-Auto	0.926	0.754	0.941	0.837	-
LTS	0.830	0.548	0.911	0.684	-

Table 2: Performance comparison of HILTS-Auto, LTS, HILTS-NN-Voting, HILTS-Random, HILTS-Uncertainty for the Animal Products use case.

LTS: This refers to the original LTS algorithm, characterized by its reliance on a manually curated validation set and no direct human label review within the iterative process for training data generation.

HILTS-Auto: This configuration represents a fully automated approach in which the validation set for monitoring performance is generated by an LLM and there is no manual label review. This setup evaluates a fully automated HILTS and how well LLM can generate labels for a classification task.

HILTS-Random, HILTS-NN-Voting, and HILTS-Uncertainty: These three instantiations represent different human-in-the-loop scenarios within the HILTS framework. In all these cases, the validation set is generated by an LLM, and human experts are involved in reviewing and correcting labels. They differ, however, in their sampling strategies for presenting examples to the human for review as explained in Section 3.4.

For the pseudo-labeling task, we utilize an open-source Large Language Model (`llama3-70b`) to generate labels for the training samples and the validation set. The base classifier, subsequently fine-tuned using the LLM-labeled data, is a text-based model (`bert-base-uncased`).

All five experiments involve an iterative active learning process that undergoes ten iterations. For the experiments involving human revision, 200 ads are pseudo-labeled by an LLM, of which 40 samples are displayed to the user in every iteration, with a total of 400 samples reviewed at the end.

5.2. Results

Use Case 1: Animal Products. Based on the results for the "Animal Products" use case shown in Table 2, HILTS-Random variant achieves the best performance with an F1-score of 0.879, driven by its leading accuracy

	Accuracy	Precision	Recall	F1	# Correct
HILTS-Random	0.850	0.812	0.853	0.832	55/400
HILTS-NN-Voting	0.730	0.665	0.766	0.712	49/400
HILTS-Uncertainty	0.784	0.690	0.922	0.790	54/400
HILTS-Auto	0.666	0.619	0.610	0.614	-
LTS	0.514	0.473	0.986	0.639	-

Table 3: Performance comparison of HILTS-Auto, LTS, HILTS-NN-Voting, HILTS-Random, HILTS-Uncertainty for Leather Products use case.

(0.946) and notably high recall (0.970), while maintaining strong precision (0.803). This indicates a highly effective reduction in false positives while capturing the most relevant ads.

In comparison, the fully automated HILTS-Auto variant achieves an F1-score of 0.837, demonstrating strong performance even without human intervention. The original LTS shows significantly lower overall performance, with an F1-score of 0.684, underscoring the substantial benefits of the algorithmic enhancements integrated within the HILTS framework.

Use Case 2: Small Leather Products. The second use case, detailed in Table 3, focuses on the "Leather Products" data collection, with LLAMA3-70b as the pseudo-labeling LLM. The HILTS-Random variant consistently achieves the highest F1-score, indicating that user verification is crucial in generating high-quality labeled data.

In comparison, the fully automated HILTS-Auto variant achieves an F1-score of 0.614, and the original LTS framework, while displaying an exceptionally high recall of 0.986 for leather products, exhibits a considerably lower precision (0.473), leading to an F1-score of 0.639.

These results for the "Leather Product" use case highlight that the HILTS framework variants, particularly those integrating human review with effective sampling strategies like random selection, significantly outperform the fully automated or original LTS approaches.

HILTS-NN-Voting vs LLM-Only. Overall, the results indicate that the algorithmic modifications implemented in the HILTS framework (e.g., changes to validation data and reward calculation, as discussed in Section 3.1) consistently produce effective downstream classifiers for the "Animal Products" and "Leather Products" tasks. This suggests that these algorithmic enhance-

ments contribute to generating higher-quality training data even before any human-in-the-loop validation or correction is applied, but are further improved when human expertise is integrated into the setting.

To better investigate the specific impact of human label correction, Figure 6 provides a more granular comparison of the HILTS-NN-Voting approach (representing a setting with human-corrected labels) and the HILTS-NN-Voting-non-corrected labels (relying solely on LLM-derived labels). For this ablation study, we examine the HILTS-NN-Voting variant since it incorporates the largest number of corrected data points (a total of 122 samples) among the HILTS framework approaches that included human intervention.

The results report the mean scores from five training runs for each batch, where we incrementally add 200 new labeled data items for each fine-tuning iteration. The cumulative number of human-corrected samples gradually increases, from four corrected samples in the first interaction to a total of 122 corrected samples at the end. In each interaction, the user reviews 40 samples.

For the full training (where the model is trained once with all 2000 samples), HILTS-NN-Voting consistently outperforms HILTS-NN-Voting-non-corrected-labels (the solid purple and blue lines in Figure 6). Specifically, the F1-score increases from 0.80 (LLM-alone baseline) to 0.85 (human-corrected). Similarly, accuracy improves from 0.90 to 0.94, and precision rises from 0.69 to 0.85. It is worth noting, however, that the LLM-alone model achieves a higher recall (0.94) for the full training set compared to the human-corrected model (0.84). This suggests that while human correction significantly enhances precision and overall F1-score, leading to a more reliable identification of the rare class, it may involve a trade-off in identifying every single instance of the rare class in this specific context.

Specialized Models vs. Foundation Models. Table 4 presents a comparison of the specialized text-based models derived from our HILTS framework (specifically, HILTS-Random, HILTS-NN-Voting, and HILTS-Uncertainty, trained using LLAMA3-70b pseudo-labels) and the few-shot classification performance of GPT-4 applied directly to the test set. It is crucial to note the substantial difference in scale between these models: GPT-4 is estimated to have parameters in the order of trillions, whereas a fine-tuned model of BERT typically has hundreds of millions of parameters, in our case, the bert-base-uncased has 110M million, making our final model significantly more resource-efficient.

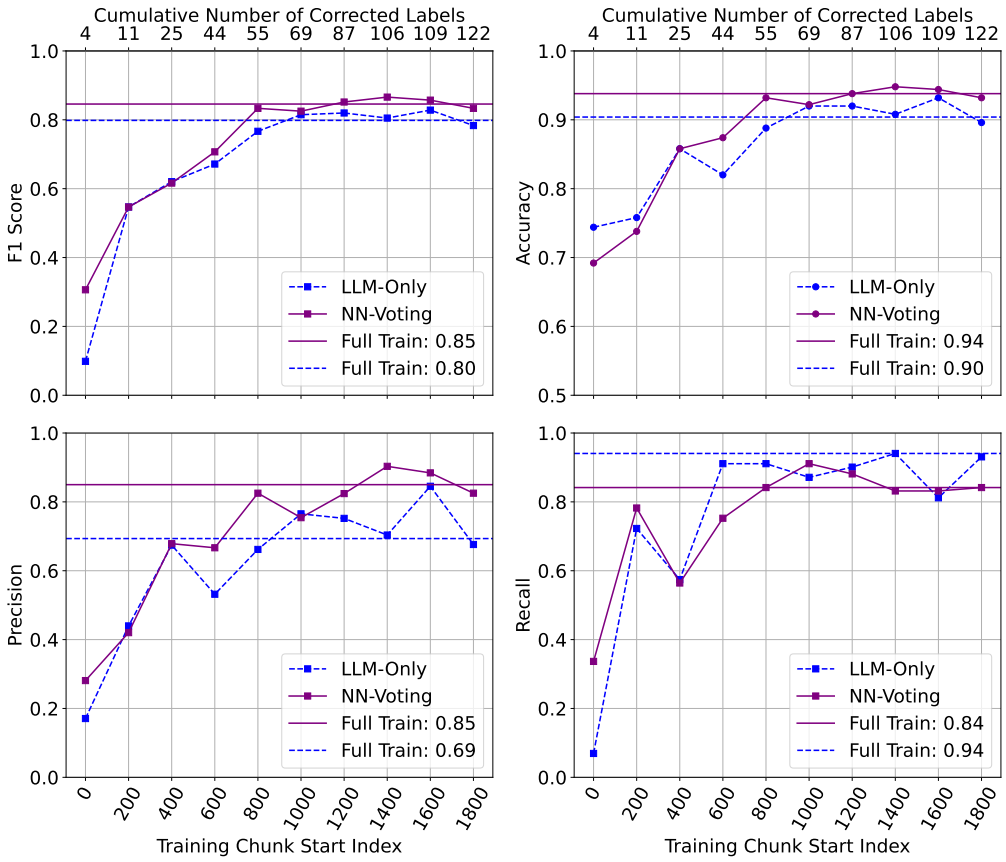


Figure 6: Metrics comparison between HILTS-NN-Voting approach vs HILTS data with no human corrected labels (LLM-Only) for models fine-tuned incrementally with 200 samples.

	Animal Products	Leather Products
GPT4	0.827	0.850
HILTS-Random	0.879	0.832
HILTS-NN-Voting	0.846	0.712
HILTS-Uncertainty	0.834	0.790

Table 4: Performance comparison (using F-1 measure) of the text-based model trained using HILTS-Random, HILTS-NN-Voting, and HILTS-Uncertainty against classification performed by GPT4. The best performance for each task is highlighted.

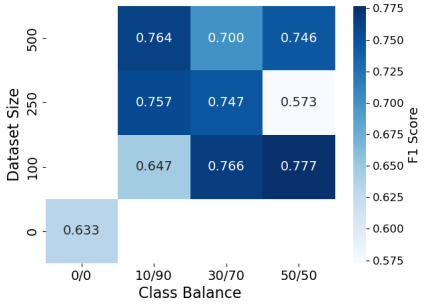
For the Animal Products task, all variants of the HILTS framework outperform direct few-shot classification by GPT-4, with the HILTS-Random variant achieving the highest F1-score of 0.879. This indicates that even with LLAMA3-70b generating the pseudo-labels, our human-in-the-loop specialized models are more effective for this specific domain.

For the Leather Products task, GPT-4 achieves a high F1-score of 0.850 in a few-shot setting, with HILTS-Random achieving a competitive F1-score of 0.832. This demonstrates that models trained using our HILTS approach, leveraging an open-source model (LLAMA) pseudo-labeling, can achieve comparable, and in the case of Animal Products, superior performance to a much larger and more expensive foundation model like GPT-4 for specific downstream tasks. This underscores the cost-effectiveness and task-specificity benefits of our framework, enabling the deployment of high-performing models at a fraction of the computational cost and parameter count of direct foundation model inference.

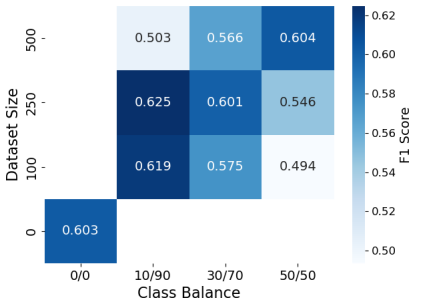
5.2.1. Analysis of Validation Set Parameters

We carried out a sensitivity analysis to assess the impact of the validation set's size and class composition on HILTS's performance. Our results, presented in Figure 7, were obtained by executing HILTS-Auto for both the animal and leather datasets. In this experiment, we fix a budget for the training data while varying other parameters. To keep costs reasonable, we've restricted this budget to 1000 labeled training samples by the LLM, representing only 50% of the amount utilized in earlier sections. With this fixed budget, we systematically varied two key properties of the validation set:

- *Validation Set Size*: We tested a range of small validation sets from 50 to 500 samples.



(a) F1 scores of different validation size/balance used on HILTS-Auto (Animal Products).



(b) F1 scores of different validation size/balance used on HILTS-Auto (Leather).

Figure 7: F1 scores of different validation size/balance used on HILTS-Auto across domains.

- *Positive Class Composition*: We adjusted the ratio of positive (rare event) to negative samples in the validation set, from highly imbalanced (10/90) to a more balanced (50/50) configuration.

We also included the scenario where no validation set was used to evaluate the performance of the model during the data sampling process. That means that for each iteration, the model trained is used in the next cycle, independently of its quality.

Our results demonstrate a positive finding regarding HILTS's robustness: As observed in figure 7a, the heatmap for the animal dataset shows that HILTS' performance (F1 score) remains consistently high and fairly stable across different validation set sizes, including the smallest ones, i.e., the dataset of size 100 had the best performance overall with 0.777 F1-score in the experiment with a 50/50 balanced ratio. Furthermore, the algorithm is not highly sensitive to the exact class composition of the validation set, maintaining strong performance even when the validation set itself is imbalanced, i.e., the dataset 500 performed second best with a class composition of 10/90. Figure 7b shows that a similar trend is observed in the leather dataset, reinforcing our findings. The model's performance shows a clear robustness against a small or incomplete validation set.

This study confirms that while a validation set is needed for metric tracking, HILTS' core strength lies in its ability to build a high-quality, large pseudo-labeled training set through intelligent active learning. Because HILTS' training process is not overly dependent on the initial quality of the validation set, it can achieve high performance even when the validation set

suffers from the same scarcity and imbalance issues present in the broader problem space.

6. Conclusion

In this paper, we introduce the *HILTS framework* and its associated system, a novel human-in-the-loop approach built upon the foundational principles of the LTS algorithm. Our core contribution lies in providing a cost-effective and scalable methodology that harnesses the power of Large Language Models (LLMs) for automated training data labeling, strategically augmented by human expertise. This synergistic combination is particularly effective for data triage tasks, especially in challenging scenarios characterized by vast, unlabeled data with highly imbalanced classes.

Our experimental evaluation on identifying "Animal Products" and "Leather Products" among large collections of ads demonstrates the superior performance of various *HILTS framework* instantiations. We show that the algorithmic modifications integrated into HILTS consistently produce high-quality labeled data for downstream classifiers, reflected in improved F1-scores, accuracy, and precision, compared to purely automated or basic LTS approaches. Our findings highlight the significant impact of integrating human correction within the labeling process– even if LLM-only validation provides a strong baseline, human feedback acts as a vital mechanism for refining data quality, mitigating LLM errors, and enhancing model robustness, particularly in identifying elusive patterns of illicit trade.

The flexible user interface of the *HILTS* allows practitioners to define specific tasks, manage validation and test datasets, and fine-tune various algorithmic parameters, including the choice of LLM and base model, as well as the sampling method for human review. We believe that *HILTS* can be a valuable and powerful tool for data scientists. Beyond combating wildlife trafficking, this adaptable framework holds immense promise for applications in other specialized domains facing similar "needle in a haystack" data challenges, such as the detection of other forms of illicit content, fraud, or rare events within large datasets. Future work will explore further optimizations for human-AI collaboration and the integration of even more complex data types.

Acknowledgments. This work was supported by NSF awards CMMI-2146306, CMMI-2146312, CMMI-2146306, and IIS-2106888. Freire was partially supported by the DARPA Automating Scientific Knowledge Extraction

and Modeling (ASKEM) program Agreement No. HR0011262087, and the ARPA-H BDF program. The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of DARPA, ARPA-H, the U.S. Government, or NSF.

References

[1] S. Chakraborty, S. N. Roberts, G. A. Petrossian, M. Sosnowski, J. Freire, J. Jacquet, Prevalence of endangered shark trophies in automated detection of the online wildlife trade, Biological Conservation 304 (2025) 110992.

[2] DARPA, Memex program, `https://www.darpa.mil/program/memex`, accessed: 2025-01-12 (2023).

[3] A. Mozer, S. Prost, An introduction to illegal wildlife trade and its effects on biodiversity and society, Forensic Science International: Animals and Environments 3 (2023) 100064.

[4] E. Demeau, M. E. V. Monroy, J. Karolan, Wildlife trafficking on the internet: a virtual market similar to drug trafficking?, Revista Criminalidad 61 (2) (2019) 101–112.

[5] B. R. Scheffers, B. F. Oliveira, I. Lamb, D. P. Edwards, Global wildlife trade across the tree of life, Science 366 (6461) (2019) 71–76.

[6] S. Nalluri, S. J. R. Kumar, M. Soni, S. Moin, K. Nikhil, A survey on identification of illegal wildlife trade, in: Proceedings of International Conference on Advances in Computer Engineering and Communication Systems: ICACECS, 2021, pp. 127–135.

[7] N. Minh, M. Willemsen, A rapid assessment of e-commerce wildlife trade in viet nam, TRAFFIC Bulletin 28 (2) (2016) 53.

[8] Q. Xu, M. Cai, T. K. Mackey, The illegal wildlife digital market: an analysis of chinese wildlife marketing and sale on facebook, Environmental conservation 47 (3) (2020) 206–212.

[9] S. Haysom, In search of cyber-enabled disruption (2019).

[10] J. Barbosa, U. Gondhali, G. Petrossian, K. Sharma, S. Chakraborty, J. Jacquet, J. Freire, A cost-effective LLM-based approach to identify wildlife trafficking in online marketplaces, in: Proceedings of the International Conference on Management of Data (SIGMOD '25), ACM, New York, NY, USA, 2025, pp. 1–14. doi:10.1145/3725256.
URL https://doi.org/10.1145/3725256

[11] J. Hernandez-Castro, D. L. Roberts, Automatic detection of potentially illegal online sales of elephant ivory via data mining, PeerJ Computer Science 1 (2015) e10.

[12] P. Siriwat, V. Nijman, Illegal pet trade on social media as an emerging impediment to the conservation of asian otters species, Journal of Asia-Pacific Biodiversity 11 (4) (2018) 469–475.

[13] E. Di Minin, C. Fink, H. Tenkanen, T. Hiippala, Machine learning for tracking illegal wildlife trade on social media, Nature ecology & evolution 2 (3) (2018) 406–407.

[14] L. Harrington, D. Macdonald, N. D'Cruze, Popularity of pet otters on youtube: evidence of an emerging trade threat, Nature Conservation 36 (2019) 17–45.

[15] R. O. Martin, C. Senni, N. C. D'Cruze, Trade in wild-sourced african grey parrots: Insights via social media, Global Ecology and Conservation 15 (2018) e00429.

[16] L. Gomez, C. R. Shepherd, Bearly on the radar–an analysis of seizures of bears in indonesia, European Journal of Wildlife Research 65 (6) (2019) 89.

[17] S. Venturini, D. L. Roberts, Disguising elephant ivory as other materials in the online trade, Tropical Conservation Science 13 (2020) 1940082920974604.

[18] D. L. Roberts, K. Mun, E. Milner-Gulland, A systematic survey of online trade: trade in saiga antelope horn on russian-language websites, Oryx 56 (3) (2022) 352–359.

[19] S. Stoner, Tigers: exploring the threat from illegal online trade, TRAFFIC Bulletin 26 (1) (2014) 26–30.

[20] S. Charity, J. M. Ferreira, Wildlife trafficking in brazil, TRAFFIC International, Cambridge, United Kingdom 140 (2020).

[21] J. Barbosa, S. Chakraborty, J. Freire, A flexible and scalable approach for collecting wildlife advertisements on the web (2024). arXiv:2407.18898.
URL https://arxiv.org/abs/2407.18898

[22] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models (2023).

[23] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training (2018).

[24] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[25] M. C. Rillig, M. Ågerstrand, M. Bi, K. A. Gould, U. Sauerland, Risks and benefits of large language models for the environment, Environmental Science & Technology 57 (9) (2023) 3464–3466.

[26] Wildlife Conservation Society, About wcs, https://www.wcs.org/about, accessed: 2024-10-08 (n.d.).

[27] N. Das, S. Chaba, R. Wu, S. Gandhi, D. H. Chau, X. Chu, Goggles: Automatic image labeling with affinity coding, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM, 2020, pp. 1717–1732.

[28] S. E. Whang, Y. Roh, H. Song, J.-G. Lee, Data collection and quality challenges in deep learning: A data-centric AI perspective, The VLDB Journal 32 (4) (2023) 791–813.

[29] G. Heo, Y. Roh, S. Hwang, D. Lee, S. E. Whang, Inspector gadget: A data programming-based labeling system for industrial images, Proceedings of the VLDB Endowment (pVLDB) 14 (1) (2020) 28 – 36.

[30] P. Varma, C. Ré, Snuba: Automating weak supervision to label training data, Proceedings of the VLDB Endowment (pVLDB) 12 (2018) 223.

[31] A. J. Ratner, S. H. Bach, H. R. Ehrenberg, C. Ré, Snorkel: Fast training set generation for information extraction, in: Proceedings of the ACM International Conference on Management of Data, ACM, 2017, pp. 1683–1686.

[32] B. B. Keskin, E. C. Griffin, J. O. Prell, B. Dilkina, A. Ferber, J. MacDonald, R. Hilend, S. Griffis, M. L. Gore, Quantitative investigation of wildlife trafficking supply chains: A review, Omega 115 (102780) (2022) 102780.

[33] A. S. Cardoso, S. Bryukhova, F. Renna, L. Reino, C. Xu, Z. Xiao, R. Correia, E. Di Minin, J. Ribeiro, A. S. Vaz, Detecting wildlife trafficking in images from online platforms: A test case using deep learning with pangolin images, Biological Conservation 279 (2023) 109905.

[34] Q. Xu, J. Li, M. Cai, T. K. Mackey, Use of machine learning to detect wildlife product promotion and sales on twitter, Frontiers in big Data 2 (2019) 28.

[35] R. Kulkarni, E. Di Minin, Towards automatic detection of wildlife trade using machine vision models, Biological Conservation 279 (2023) 109924.

[36] D.-Y. Meng, T. Li, H.-X. Li, M. Zhang, K. Tan, Z.-P. Huang, N. Li, R.-H. Wu, X.-W. Li, B.-H. Chen, et al., A method for automatic identification and separation of wildlife images using ensemble learning, Ecological Informatics 77 (2023) 102262.

[37] G. Yang, C. Sui, F. Jiang, Y. Pan, A. Zang, J. Hu, Lightweight conv-swin transformer for wildlife detection, in: 2022 International Conference on Automation, Robotics and Computer Engineering (ICARCE), IEEE, 2022, pp. 1–5.

[38] D. Chabot, S. Stapleton, C. M. Francis, Using web images to train a deep neural network to detect sparsely distributed wildlife in large volumes of remotely sensed imagery: A case study of polar bears on sea ice, Ecological Informatics 68 (2022) 101547.

[39] A. M. Roy, J. Bhaduri, T. Kumar, K. Raj, Wildect-yolo: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection, Ecological Informatics 75 (2023) 101919.

[40] S. B. Hunter, F. Mathews, J. Weeds, Using hierarchical text classification to investigate the utility of machine learning in automating online analyses of wildlife exploitation, Ecological Informatics 75 (2023) 102076.

[41] O. C. Stringham, S. Moncayo, K. G. Hill, A. Toomes, L. Mitchell, J. V. Ross, P. Cassey, Text classification to streamline online wildlife trade analyses, Plos one 16 (7) (2021) e0254007.

[42] D. Tuia, B. Kellenberger, S. Beery, B. R. Costelloe, S. Zuffi, B. Risse, A. Mathis, M. W. Mathis, F. van Langevelde, T. Burghardt, et al., Perspectives in machine learning for wildlife conservation, Nature communications 13 (1) (2022) 792.

[43] S. Rezvani, X. Wang, A broad review on class imbalance learning techniques, Applied Soft Computing 143 (2023) 110415.

[44] O. Volk, G. Singer, An adaptive cost-sensitive learning approach in neural networks to minimize local training–test class distributions mismatch, Intelligent Systems with Applications 21 (2024) 200316.

[45] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.

[46] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, J.-S. Jhang, Clustering-based undersampling in class-imbalanced data, Information Sciences 409 (2017) 17–26.

[47] A. R. Salehi, M. Khedmati, A cluster-based smote both-sampling (csbboost) ensemble algorithm for classifying imbalanced data, Scientific Reports 14 (1) (2024) 5152.

[48] H. Chamlal, H. Kamel, T. Ouaderhman, A hybrid multi-criteria meta-learner based classifier for imbalanced data, Knowledge-based systems 285 (2024) 111367.

[49] A. Gosain, S. Sardana, Handling class imbalance problem using oversampling techniques: A review, in: International conference on advances in computing, communications and informatics (ICACCI), IEEE, 2017, pp. 79–85.

[50] B. Settles, Active learning literature survey, Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009).

[51] M. Wang, X.-S. Hua, Active learning in multimedia annotation and retrieval: A survey, ACM Transactions on Intelligent Systems and Technology (TIST) 2 (2) (2011) 1–21.

[52] A. Raj, F. Bach, Convergence of uncertainty sampling for active learning, in: Proceedings of the International Conference on Machine Learning, Vol. 162, 2022, pp. 18310–18331.

[53] Y. Yang, Z. Ma, F. Nie, X. Chang, A. G. Hauptmann, Multi-class active learning by uncertainty sampling with diversity maximization, International Journal of Computer Vision 113 (2015) 113–127.

[54] R. M. Monarch, Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI, Simon and Schuster, 2021.

[55] A. Tharwat, W. Schenck, A survey on active learning: State-of-the-art, practical challenges and research directions, Mathematics 11 (4) (2023) 820.

[56] P. Lesci, A. Vlachos, Anchoral: Computationally efficient active learning for large and imbalanced datasets, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, 2024, pp. 8445–8464.

[57] C. Coleman, E. Chou, J. Katz-Samuels, S. Culatana, P. Bailis, A. C. Berg, R. Nowak, R. Sumbaly, M. Zaharia, I. Z. Yalniz, Similarity search for efficient active learning and search of rare concepts, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 6402–6410.

[58] U. Aggarwal, A. Popescu, C. Hudelot, Minority class oriented active learning for imbalanced datasets, in: International Conference on Pattern Recognition (ICPR), IEEE, online, 2021, pp. 9920–9927.

[59] Y. Xia, S. Mukherjee, Z. Xie, J. Wu, X. Li, R. Aponte, H. Lyu, J. Barrow, H. Chen, F. Dernoncourt, et al., From selection to generation: A survey of llm-based active learning, arXiv preprint arXiv:2502.11767 (2025).

[60] M. Bayer, Activellm: Large language model-based active learning for textual few-shot scenarios, in: Deep Learning in Textual Low-Data Regimes for Cybersecurity, Springer, 2025, pp. 89–112.

[61] A. H. Azeemi, I. A. Qazi, A. A. Raza, To label or not to label: Hybrid active learning for neural machine translation, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), Proceedings of the 31st International Conference on Computational Linguistics, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 3071–3082.
URL https://aclanthology.org/2025.coling-main.206/

[62] R. Xiao, Y. Dong, J. Zhao, R. Wu, M. Lin, G. Chen, H. Wang, Freeal: Towards human-free active learning in the era of large language models, arXiv preprint arXiv:2311.15614 (2023).

[63] B. Yuan, Y. Chen, Y. Zhang, W. Jiang, Hide and seek in noise labels: Noise-robust collaborative active learning with LLMs-powered assistance, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 10977–11011. doi:10.18653/v1/2024.acl-long.592.
URL https://aclanthology.org/2024.acl-long.592/

[64] M. Desmond, Z. Ashktorab, Q. Pan, C. Dugan, J. M. Johnson, Evalullm: Llm assisted evaluation of generative outputs, in: Companion Proceedings of the International Conference on Intelligent User Interfaces, ACM, 2024, p. 30–32.

[65] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, C. Zhu, G-eval: NLG evaluation using gpt-4 with better human alignment, in: "Proceedings of the Conference on Empirical Methods in Natural Language Processing", 2023, pp. "2511–2522".

[66] S. Ye, D. Kim, S. Kim, H. Hwang, S. Kim, Y. Jo, J. Thorne, J. Kim, M. Seo, Flask: Fine-grained language model evaluation based on alignment skill sets, arXiv preprint arXiv:2307.10928 (2023).

[67] C.-H. Chiang, H.-y. Lee, Can large language models be an alternative to human evaluations?, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2023, pp. 15607–15631.

[68] M. Kayali, A. Lykov, I. Fountalis, N. Vasiloglou, D. Olteanu, D. Suciu, Chorus: Foundation models for unified data discovery and exploration, Proceedings of the VLDB Endowment (pVLDB) 17 (8) (2024) 2104–2114.

[69] J. Fan, J. Tu, G. Li, P. Wang, X. Du, X. Jia, S. Gao, N. Tang, Unicorn: A unified multi-tasking matching model, SIGMOD Rec. 53 (1) (2024) 44–53. doi:10.1145/3665252.3665263.
URL https://doi.org/10.1145/3665252.3665263

[70] Z. Tan, D. Li, S. Wang, A. Beigi, B. Jiang, A. Bhattacharjee, M. Karami, J. Li, L. Cheng, H. Liu, Large language models for data annotation and synthesis: A survey, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 930–957.

[71] X. Wang, H. Kim, S. Rahman, K. Mitra, Z. Miao, Human-llm collaborative annotation through effective verification of llm labels, in: Proceedings of the CHI Conference on Human Factors in Computing Systems, 2024, pp. 1–21.

[72] Z. Zhu, Y. Wang, S. Yang, L. Long, R. Wu, X. Tang, J. Zhao, H. Wang, Coral: Collaborative automatic labeling system based on large language models, in: International Conference on Very Large Databases (VLDB-Demo), VLDB Endowment, 2024, pp. 4401–4402.

[73] L. Weber, B. Plank, Activeaed: A human in the loop improves annotation error detection, in: Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, 2023, pp. 8834–8845.

[74] N. Pangakis, S. Wolken, Keeping humans in the loop: Human-centered automated annotation with generative AI, arXiv preprint arXiv:2409.09467 (2024).

[75] V. Satopaa, J. Albrecht, D. Irwin, B. Raghavan, Finding a" kneedle" in a haystack: Detecting knee points in system behavior, in: 2011 31st international conference on distributed computing systems workshops, IEEE, 2011, pp. 166–171.

[76] S. Bhardwaz, R. Godha, Svelte. js: The most loved framework today, in: 2023 2nd International Conference for Innovation in Technology (INOCON), IEEE, 2023, pp. 1–7.

[77] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International conference on machine learning, PMLR, 2021, pp. 8748–8763.

[78] Z. Li, X. Wu, H. Du, F. Liu, H. Nghiem, G. Shi, A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges, arXiv preprint arXiv:2501.02189 (2025).

[79] LanceDB contributors, LanceDB: A database for vector-search with Lance format, `https://lancedb.github.io/lancedb/`, accessed: 2025-06-16 (2024).

[80] VIDA-NYU, Ache crawler, `https://github.com/VIDA-NYU/ache`, accessed: 2025-04-09 (2025).

[81] C. Secretariat, World wildlife trade report 2022, Tech. rep., Secretariat of the Convention on International Trade in Endangered Species of Wild Fauna and Flora (CITES), Geneva, Switzerland (2022).

[82] O. C. Stringham, S. Moncayo, E. Thomas, S. Heinrich, A. Toomes, J. Maher, K. G. Hill, L. Mitchell, J. V. Ross, C. R. Shepherd, et al.,

Dataset of seized wildlife and their intended uses, Data in Brief 39 (2021) 107531.

[83] eBay Inc., ebay: Buy, sell, and save on brands you love, https://www.ebay.com, accessed: 2024-10-14 (2023).