LLM-in-the-Loop: Replicating Human Insight with Instruction-tuned LLMs

Mengze Hong [1] Wailing Ng [1] Di Jiang [2] Chen Jason Zhang [1] Lei Chen [3]

Abstract

Building on the success of human-in-the-loop, where human wisdom is integrated into the development of machine learning algorithms, we take the initiative to envision an innovative and promising paradigm, **LLM-in-the-loop (LLM-ITL)**, which leverages the unique advantages of LLMs to replicate human involvement and offer a more flexible and cost-efficient solution to real-world challenges. Through a comprehensive review of LLM research from 2020 to 2024, we reveal that many existing LLM applications inherently align with LLM-ITL, with researchers rapidly claiming their superiority over machine learning baselines and LLM-native solutions; however, no universal definition exists, hindering its further advancement and application. In this paper, we define and categorize LLM-ITL methodologies for data, model, and task-centric applications, discuss their underlying rationale, and highlight emerging areas where LLMs can be further integrated into the loop. Furthermore, we present opportunities for developing better LLM-ITL solutions with technical advancements, such as LLM crowdsourcing and text-to-solution, establishing the proposed paradigm as a promising avenue for the future of LLM applications and machine learning research.

1. Introduction

Human-in-the-loop has gained increasing popularity for solving real-world problems by integrating human knowledge and expertise into the development of machine learning models (Wu et al., 2022; Fang et al., 2023). With the recent emergence of Large Language Models (LLMs) and their products, such as ChatGPT and Claude, many researchers argue that LLMs not only significantly outperform traditional machine learning baselines, but also surpass human experts in many tasks (Pu et al., 2023; Törnberg, 2023; Gilardi et al., 2023). As LLMs evolve to become more agent-like and with the proven effectiveness of the “in-the-loop” techniques, a novel application paradigm, “**LLM-in-the-loop**” (abbreviated as **LLM-ITL**), has emerged as a focal point of interest for both academia and industry.

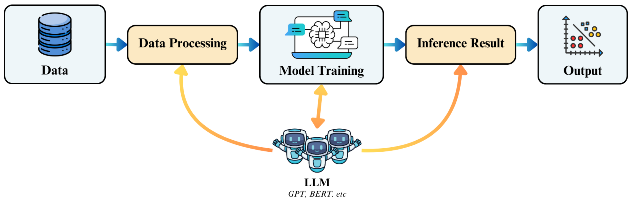


Figure 1. Overview: LLM-in-the-loop model development

The term “LLM-in-the-loop” has gained considerable attention due to the expanding capabilities and popularity of LLMs, yet no universal definition exists in the current research landscape. Interpretations vary from narrowly defining it for specific tasks or methods (Yang et al., 2024b; Kholodna et al., 2024) to adopting an overly broad scope that might generalize the concept (Sudhakar et al., 2024; Zhang et al., 2024b; Bartolo et al., 2020). This concept has also become a catchphrase to align with LLM application trends (Wu et al., 2024; Keles et al., 2024), leading to ambiguity and confusion. Appendix A presents a comprehensive list of research papers collected up to December 31, 2024, featuring the keywords LLM-in-the-loop” or LLM-ITL” in their titles or abstracts. These examples underscore the increasing interest in LLM-ITL applications. However, without a clear definition, there is a lack of understanding of how to effectively utilize LLMs. This lack of clarity limits their generalizability and hinders the recognition of their broader potential to enhance various stages of the problem-solving pipeline, ultimately overlooking their unique contributions to the field of machine learning.

In this position paper, we envision the future of LLM applications and argue that the “LLM-in-the-loop” paradigm, which harnesses the strengths of both LLMs and traditional machine learning algorithms, presents prevailing research opportunities and practical benefits. Through comprehensive literature reviews and detailed case study analyses, we demonstrate the growing popularity and effectiveness of this

[1] The Hong Kong Polytechnic University [2] AI Group, WeBank Co., Ltd. [3] The Hong Kong University of Science and Technology (Guangzhou). Correspondence to: Di Jiang <dijiang@webank.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

framework, as evidenced by the widespread, although often unconscious, application of its methodologies and the resulting state-of-the-art performances. However, this increased visibility also highlights concerns about a lack of public understanding, motivating our efforts to define the framework formally, provide best practices for its application, and offer insights for future research directions.

Contribution. This paper pioneers a detailed discussion on understanding LLM-in-the-loop machine learning, establishing it as a promising framework for the future of LLM applications in solving real-world problems. The contributions include: 1) We present practical scenarios where directly applying LLMs for problem-solving results in suboptimal outcomes, highlighting the importance of integrating conventional machine learning algorithms in the era of LLMs; 2) By synthesizing insights from related concepts and examining the implementation of existing in-the-loop methodologies, we formulate the LLM-in-the-loop framework from three perspectives, providing a foundation for future research; 3) We identify challenges in developing effective LLM-in-the-loop solutions and present promising avenues for future research and impactful applications, guiding the research community towards an underexplored landscape of LLM application and machine learning research.[1]

2. Background

2.1. Large Language Model: Trends and Challenges

LLM Applications. Among diverse areas of LLM research, the study of "Applications of LLMs/ChatGPT" has emerged as the fastest-growing topic since 2023 (Movva et al., 2024). These applications increasingly adopt a **Model-as-a-Service (MaaS) paradigm** (Sun et al., 2022; Gan et al., 2023), also referred to as LLM-native solutions (Liang et al., 2024), which deliver a broad range of functionalities through easily accessible interfaces. As generative language models, LLMs excel in tasks that are inherently sequence-to-sequence (seq2seq) (Vaswani et al., 2017), such as natural language comprehension, translation, and generation (Sottana et al., 2023; Bahdanau et al., 2015; Sutskever et al., 2014; Lewis et al., 2020a). However, extending their application to real-world problem-solving presents significant challenges (Chen et al., 2025a), as these tasks often diverge from the fundamental nature of language modeling and extend beyond the scope of NLP (Srivatsa & Kochmar, 2024; Chen et al., 2024d). Even for tasks that appear NLP-relevant, such as text clustering and topic modeling, the underlying processes do not naturally conform to a seq2seq setting, often relying more on representation learning and optimization rather than generative capabilities (Bengio et al., 2013).

While much of the application-driven research advancements focus on developing better LLMs and innovative engineering techniques (Chen et al., 2023), such as prompt engineering (Song et al., 2024; Brown et al., 2020), model finetuning (Hu et al., 2022), and Retrieval Augmented Generation (Lewis et al., 2020b), commendable research efforts are also being made to explore the use of existing state-of-the-art LLMs or smaller, more cost-efficient models (Xu et al., 2024), within **better-designed problem-solving workflows**, such as LLM-chaining (Grunde-McLaughlin et al., 2024) and multi-agent collaboration (Hong et al., 2024c). Task decomposition techniques have further emerged as a promising solution for complex, multi-step tasks (e.g., planning a wedding) (Yuan et al., 2025; Huang et al., 2023), where prompting-based LLMs and machine learning algorithms collaborate effectively in solving well-structured sub-tasks (Khot et al., 2023).

LLM vs. Human. With LLMs demonstrating increasing capabilities across various benchmark evaluations, especially when provided with clear instructions and demonstrations, He et al. (2024) pose a critical and significant inquiry: **Can LLMs potentially replace crowdsourced annotators?** Törnberg (2023) finds that GPT-4 achieves higher accuracy, greater reliability, and equal or lower bias than human classifiers when given the same instructions for tweet classification. This emphasizes the relatively low technical requirement of deploying LLM, as the instructions initially provided to human workers can be reused. Similarly, Gilardi et al. (2023) demonstrates that zero-shot GPT-3.5 outperforms certified "MTurk Masters" high-ability crowd workers in text-annotation tasks. Cegin et al. (2023) suggests that ChatGPT can perform data augmentation with greater lexical and syntactic diversity than human workers, resulting in reliable downstream performance where models trained on ChatGPT-generated data exhibiting comparable robustness to those trained on data from human crowds. With the comparable performance, the resource efficiency of LLM demonstrates substantial advantages. Gilardi et al. (2023) reveal that employing an LLM for data labeling is cost-effective, with the per-annotation cost of ChatGPT being 30 times cheaper than MTurk. Additionally, Cegin et al. (2023) claims that substituting human workers with LLMs for generating new data instances is 600 times cheaper.

Incapabilities of LLM. While LLMs excel in numerous tasks, practical scenarios exist where they either underperform or prove infeasible compared to traditional machine learning methods (Liu et al., 2024b). One prominent issue emerges as the observation of misalignment between the number of input instances and the corresponding labels generated by LLMs, initially described as "omission error" in data annotation task (Kholodna et al., 2024). However, this observation remains largely unexplored by the research

[1] Code and reading list are available at GitHub repository.

community due to a lack of clear problem formulation. We argue that the misbehavior of LLM is largely due to the absence of a hard-coded solution space, which is often weakly specified through instruction prompts (Zeng et al., 2024), unlike traditional machine learning that strictly binds the solution space and model behavior. To formally define this limitation, we formulate a problem abstraction as follows:

Definition 2.1. Given input data $\mathcal{D}$, targeted solution space $\mathcal{S}$, and an instruction prompt $\mathcal{P}(\mathcal{S})$ specifying the solution space, the failure occurs when:

$$LLM(\mathcal{P}(\mathcal{S}), \mathcal{D}) \subseteq \mathcal{R}$$
$$\text{where} \quad ||\mathcal{R} - \mathcal{S}||^2 > \epsilon$$

where the generated result space deviates significantly from the targeted solution space, exceeding a threshold ϵ.

2.2. In-the-loop Solutions

Human-in-the-loop. Human-in-the-loop (HITL) is a well-established approach for incorporating human expertise (Agarwal et al., 2023) into automated modeling processes to enhance the accuracy of predictive models (Kumar et al., 2019), with proven performance improvement and enhanced interpretability in various tasks such as sentence parsing (He et al., 2016), topic modeling (Kumar et al., 2019), and text classification (Arous et al., 2021). Extensive research efforts have explored HITL workflows in machine learning, focusing on data preprocessing, model training, and system-independent application (Wu et al., 2022). Moreover, HITL is particularly beneficial when machine learning models encounter difficulties with complex, nuanced, or ambiguous tasks that demand prior knowledge (Diligenti et al., 2017) and contextual understanding (Mosqueira-Rey et al., 2022).

Definition of LLM-in-the-loop. Drawing inspiration from the close relationship with human-in-the-loop, the LLM-in-the-loop paradigm is defined as the integration of LLM interaction, intervention, and judgment to guide or modify the training and inference processes of a machine learning model. While it mirrors the human-in-the-loop process by substituting human participation with LLM agents, **the inference remains the responsibility of the machine learning model rather than the LLM agent** (or human worker, as in HITL), distinguishing it from LLM-native or LLM-ML collaboration where the LLM plays the central role. Notably, given the widespread availability and scalability of LLM agents compared to human workers, we argue that **LLM-in-the-loop offers broader applicability across training, inference, and deployment stages, positioning it as a more general framework that encompasses and extends existing in-the-loop methodologies**. In the following discussion, we demonstrate how LLMs can effectively replace the human role and provide additional benefits to the development of machine learning algorithms.

3. Understanding LLM-in-the-loop

3.1. Case Study: LLM-in-the-loop Clustering

Human-in-the-loop methodologies have been extensively applied in clustering problems to integrate prior knowledge into unsupervised learning (Coden et al., 2017; Srivastava et al., 2016; Holzinger, 2016). Recently, the development of LLM-in-the-loop solutions for text clustering has rapidly emerged, achieving state-of-the-art performance by leveraging the language understanding capabilities of LLMs.

Observation and Motivation. The research community appears inherently aware of the limitations in directly applying LLMs for text clustering, as evidenced by the observation that existing studies rarely consider LLM-native baselines but compare solely with conventional machine learning algorithms when developing LLM-ITL solutions (Viswanathan et al., 2024; Hong et al., 2024a; Pattnaik et al., 2024; Zhang et al., 2023b). To fill in the gap of missing LLM-native results, we present an empirical study in Appendix B. Notably, the clustering problem has a strict solution space defined by n instances k candidate labels. Our findings reveal that over 90% of the LLM-generated results fail to capture the targeted number of labels and are misaligned with the input instances. Both the instruction prompt and input data affect inference behavior, yet the problem remains unsolved even with state-of-the-art prompt tuning technique (Agarwal et al., 2024) and in simple clustering settings. This observation can be further generalized into the following abstraction:

Definition 3.1. Given an instruction prompt $\mathcal{P}(n \rightarrow n)$ with n input instances $\{d_1, \dots, d_n\}$ and a predefined candidate space $\mathcal{S}$, the failure occurs when:

$$LLM(\mathcal{P}, \{d_1, \dots, d_n\}) \rightarrow \{r_1, \dots, r_m\} \subseteq \mathcal{S}'$$
$$\text{where} \quad m \neq n \quad \text{and/or} \quad \exists s' \in S',\ s' \notin \mathcal{S}$$

where the number of outputs m deviates from the expected n, and/or the space of generated label $\mathcal{S}'$ contains at least one unexpected label s' that deviates from $\mathcal{S}$.

This motivates the development of LLM-in-the-loop solutions that rely on machine learning algorithms to produce cluster assignments under the targeted solution space.

LLM-in-the-loop Solutions. ClusterLLM represents a pioneering LLM-in-the-loop solution for text clustering (Zhang et al., 2023b), addressing the limitations of LLM-native approaches in having restricted access to embedding vectors. API-based LLM is prompted to respond to pairwise preference questions structured as a triplet, consisting of two candidate instances and a reference anchor. These preferences are used to fine-tune an embedder, ensuring the input corpus is mapped to a refined embedding space for better clustering. This outlines a typical in-the-loop methodology where **the input data is preprocessed before the modeling process**. For instance, Viswanathan et al. (2024) augmented the input data through a keyphrase expansion strategy, generating a set of keyphrases that could describe document intent with LLM. The sentence and keyphrase embeddings are then concatenated to create a task-dependent data representation for better intent clustering. Similarly, Pattnaik et al. (2024) prompted a fine-tuned LLM to generate a concise cluster name and description for each cluster, then combining these embeddings with the cluster centroid embedding to create weighted multi-view representations, enhancing the performance of the agglomerative clustering algorithm in deriving topical categories within the documents.

Besides incorporating LLMs into the data preprocessing phase, Hong et al. (2024a) proposed the idea of iterative clustering with LLMs feedback, where initial cluster assignments obtained from K-means are evaluated by fine-tuned LLM based on semantic coherence, and the poorly formed clusters are refined to enhance the final result. Similarly, Viswanathan et al. (2024) prompted LLM to select data instances that *must* be linked or *cannot* be linked, forming a pairwise constraint clustering with the PCKMeans algorithm. These approaches transform the original nature of unsupervised learning into an interactive or semi-supervised learning process, embodying a philosophy of designing LLM-in-the-loop solutions that **modify the modeling process with LLM-driven utilities**.

Furthermore, developing task-specific applications requires a task-oriented design. In the intent clustering problem, Hong et al. (2024a) proposed using LLMs to name clusters in the "action-objective" form, which enhances the usability of the clustered results and allows for further refinement based on either the action or the objective. Likewise, Viswanathan et al. (2024) utilized the reasoning capability of LLMs to assess whether a given low-confidence point belongs to the current cluster, performing post-correction on relocating the data point based on the LLM's judgment. These methods enable **further refinement of the modeling results with system-independent LLM utilities**.

Based on the case study of LLM-in-the-loop solutions in text clustering, the methodologies can be categorized according to the specific purposes of LLM integration, namely: **data-centric, model-centric, or task-centric**. This framework enables a comprehensive exploration of the associated techniques and highlights opportunities for applying LLM-in-the-loop methods in underutilized domains.

3.2. LLM-in-the-loop: Data-Centric Approaches

The data-centric approach employs LLMs during the data preprocessing stage of machine learning modeling, with the goal of improving data quality, diversity, and representation to facilitate effective model training and address challenges inherent in traditional data preparation workflows.

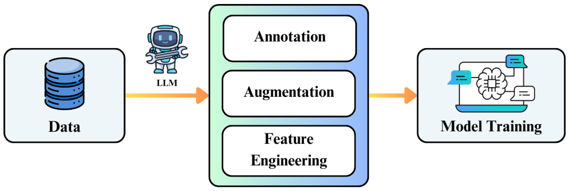


Figure 2. Overview: Data-centric LLM-in-the-loop

Definition 3.2. Given an original dataset $\mathcal{D}_0$, learning function F, and a LLM-driven transformation function Φ_{LLM} guided by prompt $\mathcal{P}$, the data-centric approach aims to improve the task-specific loss $\mathcal{L}$ through data enhancement:

$$\begin{aligned} \text{Preprocess:} \quad & \mathcal{D}_{\text{tf}} = \Phi_{\text{LLM}}(\mathcal{D}_0, \mathcal{P}), \\ \text{Train:} \quad & M_{\text{tf}} = F(\mathcal{D}_{\text{tf}}), \\ \text{Target:} \quad & \mathcal{L}(M_{\text{tf}}) < \mathcal{L}(M_0) \end{aligned} \tag{1}$$

where the preprocessed dataset $\mathcal{D}_{\text{tf}}$ enables the training of model M_{tf} to achieve superior performance compared to the model trained on the original dataset, denoted as M_0.

Data Annotation. Data annotation is a fundamental step in supervised machine learning; however, the annotation process is labor-intensive and often suffers from inconsistent quality due to inherent biases and a lack of expertise (Pandey et al., 2022; Hettiachchi et al., 2021). Recent advancements demonstrate the potential of LLMs to revolutionize this process by offering efficient, high-quality, and scalable annotation solutions (Tan et al., 2024), often matching or exceeding the quality achieved by crowdsourced annotators and domain experts (Gilardi et al., 2023; Kuzman et al., 2023; Törnberg, 2023). For instance, Chen et al. (2024c) showcased their effectiveness in event extraction, and Kuzman et al. (2023) highlighted ChatGPT's superior performance in automatic genre identification on unseen datasets. Innovative strategies, such as Chain-of-Thought (CoT) prompting combined with explain-then-annotate workflows (He et al., 2024), and CoT with majority voting (Choi et al., 2024), have further advanced LLM-based annotation methods, enabling human-like precision in complex tasks. Moreover, Smith et al. (2024) introduced the concept of Prompted Weak Supervision, which leverages LLMs to generate probabilistic labels, reducing the need for manual intervention while maintaining high annotation quality.

Data Augmentation. Data augmentation is a critical yet complex task that goes beyond basic labeling, requiring the generation of diverse fundamental and auxiliary information tailored to specific task requirements (Rebuffi et al., 2021; Hong et al., 2024b). Although crowdsourcing can be used to address this need, producing reliable and high-quality augmented data poses a far greater challenge than data annotation, whereas conventional generative models also fall short of meeting these demands (Yang et al., 2023). In this context, LLMs present a transformative solution by generating diverse, contextually enriched synthetic datasets, significantly reducing the dependence on manual data collection. For example, Yu et al. (2024) introduced the use of attributed prompts to generate attribute-specific synthetic data, while Zou et al. (2024) proposed a collaborative framework utilizing multiple LLMs to create high-quality synthetic datasets. In addition, Choi et al. (2024) demonstrated the capability of LLMs to create domain-agnostic datasets, paving the way for universal domain generalization. Ba et al. (2024) also illustrated how synthetic data generation with LLMs can reduce calibration errors and improve accuracy on real-world test datasets.

Feature Engineering. Feature engineering transforms raw data into interpretable representations that enhance model performance (Hollmann et al., 2024). Traditional methods rely primarily on domain expertise, but the combinatorial complexity of manually exploring feature spaces renders this approach impractical (Gu et al., 2024). Recent advances leverage LLMs to automate and refine feature generation, producing semantically rich, context-aware features aligned with dataset characteristics and task objectives. For instance, Zhang et al. (2024c) introduced an LLM-driven framework for iterative feature generation and performance-guided refinement. Balek et al. (2024) further demonstrated that LLMs generate interpretable textual features surpassing traditional representations like bag-of-words or dense embeddings in discriminative power. Beyond text, LLMs can align diverse representations for structured learning tasks, such as converting environmental data into structured domain-specific language for agent learning (Spiegel et al., 2024) or encoding conversational turns into canonical forms to support domain-general dialogue policies (Sreedhar et al., 2024). Furthermore, Yang et al. (2024a) emphasized LLMs' versatility to generate task-relevant, linguistically grounded features, such as extracting subject-object pairs.

Discussion. The integration of LLMs into data preprocessing offers undeniable advantages in mitigating labor-intensive workflows, and the research question of **how to make LLMs better data annotators** represents a prominent research direction combining LLMs and data science. However, the development of in-the-loop solutions poses new challenges, requiring both model-specific adaptations (e.g., augmenting data embeddings to fit the particular optimization mechanism) and task-specific customizations (e.g., crafting specific features for intended purposes). This introduces a high level of diversity in how data can be enhanced. While LLMs demonstrate emerging capabilities with in-context learning and can provide domain-specific knowledge often lacking in machine learning, the exploration of applying LLMs in a typical in-the-loop solution to fully leverage these capabilities remains largely underexplored, with limited research combining LLM-driven data preprocessors and machine learning models to solve real-world problems. Additionally, the use of LLMs poses new concerns in assessing data integrity and detecting underlying biases and false information caused by potential hallucinated generations (Tan et al., 2024), thereby motivating further research into both in-the-loop design and LLM-native techniques to mitigate these inherent limitations.

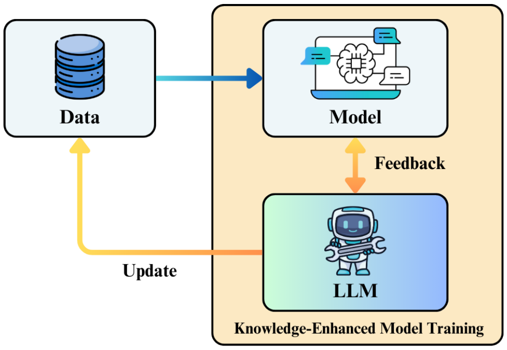


Figure 3. Overview: Model-centric LLM-in-the-loop

Our position: *From a data-centric perspective, LLM-in-the-loop benefits model training by alleviating data scarcity and enriching data features. The integration of LLMs in a crowdsourcing-like fashion has proven particularly effective, providing a valuable framework for developing "labor-free" in-the-loop solutions. Future research should focus on 1) identifying innovative ways to integrate prior knowledge from LLMs into data features and 2) designing robust crowdsourcing approaches with LLM agent collaboration. These advancements hold the potential to significantly address the long-standing challenges of data availability and quality assurance.*

3.3. LLM-in-the-loop: Model-Centric Approaches

Incorporating rich human knowledge into machine learning models has been a longstanding research focus, as machine learning alone cannot fully capture the depth of human domain expertise (Wu et al., 2022). To address this, human-in-the-loop approaches integrate human insights by iteratively refining the model for knowledge-enhanced learning. In this section, we explore how LLMs can substitute for the human role to provide model-centric support.

Definition 3.3. Given a trained machine learning model M and LLM-driven utility Φ_{LLM} guided by instruction prompt $\mathcal{P}$, the model-centric approach aims to improve the task-specific loss $\mathcal{L}$ through model refinements:

$$\begin{aligned} \text{Refine:} \quad & M_{\text{tf}} = \Phi_{\text{LLM}}(M, \mathcal{P}), \\ \text{Target:} \quad & \mathcal{L}(M_{\text{tf}}) < \mathcal{L}(M) \end{aligned} \tag{2}$$

such that the refined model M_{tf} outperforms the original model M.

Active Learning and Iterative Refinement. Active learning is a crucial technique for integrating human wisdom and prior knowledge into iterative learning frameworks, especially in low-resource learning settings (Zhang et al., 2023a). Recently, there has been a growing interest in leveraging LLMs for both annotation and uncertainty estimation in an integrated active learning setting across various NLP tasks, such as text classification (Rouzegar & Makrehchi, 2024), named entity recognition, and relation extraction (Zhang et al., 2023a). Unlike data augmentation with LLMs, active learning is a model-based approach that focuses on uncertainty sampling - selecting data points where the model is most uncertain, thus allowing it to learn from challenging instances (Rouzegar & Makrehchi, 2024). While sample selection can be complex and necessitates human judgment, the concept of LLM confidence estimation offers a valuable alternative (Xiong et al., 2024; Geng et al., 2024), enabling verbalized confidence scores to assist the sampling process.

Beyond direct annotation, LLMs also provide a feedback mechanism in an iterative setting, addressing limitations in tasks where direct annotation is challenging (e.g., clustering). For instance, An et al. (2024) queried LLMs to identify true neighbors of selected samples from multiple candidates, leveraging this information for contrastive learning to improve base model representation. Similarly, Hong et al. (2024a) employed LLMs to iteratively refine poorly formed clusters through coherence evaluation at each iteration. In topic modeling, Yang et al. (2024b) used LLMs to refine topics generated by the base model, aligning the model with LLM-provided refinements through fine-tuning. These applications share the commonality of involving LLMs not only in the model training process but also in the inference and deployment stages, as most discussed applications pertain to unsupervised learning. This underscores another unique advantage of LLM-in-the-loop: its inherent model-in-the-loop nature, which offers deployment flexibility and facilitates application across diverse scenarios.

Reinforcement Learning. Reinforcement learning (RL) is a crucial segment of machine learning that seeks to align model behaviors with human expectations through a feedback mechanism (Cao et al., 2024). As LLM agents are increasingly calibrated to human behaviors and preferences through alignment techniques (Liu et al., 2024a; Wang et al., 2023), LLM-in-the-loop reinforcement learning has gained significant momentum. Existing research suggests that the prior knowledge of LLMs can be integrated into the RL process by serving as dynamic feedback sources, such as natural language instructions, demonstrations, evaluative signals, and informative guidance (Laleh & Ahmadabadi, 2024). For instance, Du et al. (2023) leveraged pre-trained LLMs to provide intrinsic motivation for RL agents by setting exploration goals and issuing rewards upon their completion. Similarly, Kwon & Michael (2023) employed LLMs as reward functions, where agent behaviors are evaluated against desired outcomes, generating corresponding reward signals. Barj & Sautory (2024) utilized LLM feedback to refine RL policies, particularly in scenarios where agents struggled to generalize to out-of-distribution environments.

In addition to reward setting, Karimpanal et al. (2023) utilized LLMs to generate decision-making behaviors, thereby accelerating the learning process. Similarly, Prakash et al. (2023) guided agent exploration by evaluating actions and behaviors based on observed states and task descriptions. In scenarios where RL agents need access to confidential information, Moradi et al. (2023) proposed integrating human-in-the-loop with Federated Learning. However, human involvement may still compromise data privacy and increase the cost of preventive measures. By introducing LLM-in-the-loop with locally deployed open-source LLMs, data privacy can be significantly enhanced, ensuring compliance with the principle of "keeping original data within the domain and making data available and invisible" (Yang et al., 2019). This approach further highlights the unique advantage of having a (large language) model-in-the-loop in constrained scenarios where human involvement is not preferred.

Our position: *LLMs demonstrate transformative potential in supporting knowledge-enhanced machine learning with iterative updating. They offer scalable and cost-efficient alternatives to traditional human involvement, facilitating deployable solutions due to their automated nature. However, the limitations of LLMs can be amplified by their direct interaction with the modeling process, leading to issues such as 1) poorly calibrated LLMs generating biased feedback and 2) failures in data sampling and labeling that create outliers in the iterative refinement process. These issues are difficult for machine learning models to unlearn and are hard to detect, unlike errors in data preprocessing.*

3.4. LLM-in-the-loop: Task-Centric Approaches

The task-centric approach employs LLMs as versatile and powerful utilities tailored for specific tasks or applications, focusing on enhancing task performance (e.g., prediction accuracy and interpretability). This section examines how LLMs can be strategically integrated into the inference and post-inference stages of problem-solving.

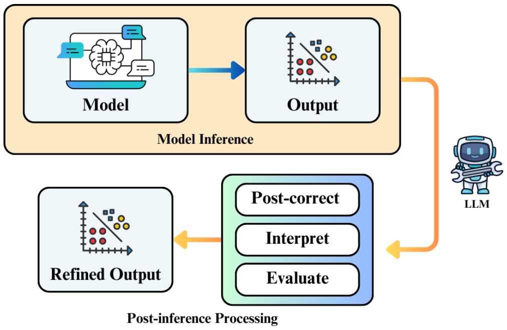


Figure 4. Overview: Task-centric LLM-in-the-loop

Definition 3.4. Given a trained machine learning model M, inference task T, and LLM-driven utility Φ_{LLM} guided by prompts $\mathcal{P}$, the task-centric approach aims to enhance task-specific performance evaluation U (e.g., accuracy, coherence) by incorporating LLMs during inference or post-inference evaluation stage:

$$\begin{aligned} \text{Inference:} \quad & \mathcal{O} = M(T), \\ \text{Support:} \quad & \mathcal{O}^* = \Phi_{\text{LLM}}(M, \mathcal{O}, \mathcal{P}) \\ \text{Target:} \quad & U(\mathcal{O}^*) > U(\mathcal{O}) \end{aligned} \tag{3}$$

where the LLM integration interacts with the model output and provides task-specific support, such as post-correction and explainability enhancement.

Post-Correction. Post-correction aims to improve machine learning predictions after the training process by refining model outputs with minimal local changes, a task where traditional methods often fall short due to their limited contextual understanding and scalability (Wei et al., 2024). With the extensive pre-trained knowledge in LLMs, Zhong et al. (2024) proposed using LLMs with in-context learning as post-hoc correctors to propose corrections for the predictions of machine learning models, enabling them to integrate contextual knowledge and deliver dynamic, context-aware corrections. In automatic speech recognition (ASR), CHEN et al. (2023) demonstrated the utility of LLMs in leveraging N-best hypothesis lists to predict the final output and found that LLM can correct errors even for tokens absent from the hypothesis list. Similarly, Hu et al. (2024) employed LLMs to synthesize diverse translation outputs from multiple N-best hypotheses, resulting in a substantial enhancement in translation quality. Beyond ASR, LLMs have been applied in clustering, where Viswanathan et al. (2024) re-ranked low-confidence points by querying their correctness against representative points, and Hong et al. (2024a) refined clusters by generating descriptive names and summaries using LLMs. In topic modeling, Chang et al. (2024) used LLMs to iteratively refine topics by identifying misaligned terms and replacing them with contextually appropriate alternatives.

Model Interpretability. Machine learning models frequently struggle with interpretability, especially when generating natural language explanations or extracting actionable insights from outputs. Conventional techniques like feature importance scores and attribution maps focus on explaining model decisions but lack the capacity to interpret outputs through human-intuitive narratives (Pang et al., 2024). LLMs mitigate this gap by synthesizing their natural language understanding and generative capabilities to contextualize model outputs. For instance, Pattnaik et al. (2024) employed LLMs to generate descriptive cluster labels and summaries, while Hong et al. (2024a) and An et al. (2024) assign semantically meaningful names to clusters. In social media analysis, Islam & Goldwasser (2024) leveraged LLMs to summarize high-impact instances within clusters, producing cohesive "talking points" that directly supported downstream tasks like stance detection and demographic inference. Liu et al. (2023) explored the application of LLMs in evaluating text quality and open-ended responses, providing enriched insights by extracting additional features for metric evaluation. Additionally, Bhattacharjee et al. (2024) enabled causal explainability via LLMs by generating counterfactual explanations in black-box text classifiers, enhancing interpretability across complex ML workflows.

Discussion. Traditional inference workflows often underutilize intermediate outputs, such as hypotheses, embeddings, or raw predictions, leaving valuable information unexplored. Rule-based or heuristic post-processing methods lack the adaptability and contextual understanding needed to handle complex or ambiguous scenarios effectively (CHEN et al., 2023). Similarly, traditional interpretability techniques, such as feature importance scores or attribution maps, provide limited insights and fail to produce human-interpretable explanations or actionable feedback (Zytek et al., 2024). LLMs address these limitations by leveraging extensive pre-trained knowledge and few-shot capabilities to dynamically refine outputs, aligning them with task-specific requirements (Viswanathan et al., 2024). Moreover, LLMs can generate high-level abstractions, such as descriptive summaries (Pattnaik et al., 2024) and novel metrics (Liu et al., 2023), surpassing the rigid constraints of conventional approaches and enabling more flexible insights.

Alternative Views. In these discussions, LLM-driven utilities are designed to facilitate in-the-loop development, essentially serving as **LLM-native components** tailored for sub-tasks (e.g., evaluation, annotation) rather than solving the entire problem. While we acknowledge the limitations of LLM-native applications in constrained problem-solving scenarios, they remain a feasible and predominant choice for many less-restricted tasks, such as code generation and machine translation. LLM-native solutions are particularly well-suited for tasks involving multiple input sources and modalities (Tang et al., 2024), complex reasoning (Ahn et al., 2024), and heavy reliance on domain knowledge (Bi et al., 2024). These are areas where traditional machine learning algorithms, even with human or LLM in-the-loop, struggle to perform effectively, highlighting the need for LLM-native solutions to be developed and applied.

Our position: *Designing better task-centric LLM-ITL solutions is becoming a scientific endeavor, presenting numerous new challenges and research opportunities. These include 1) replicating human-in-the-loop strategies while adapting to the unique characteristics of LLMs and 2) innovating LLM techniques to enhance their involvement in task-centric applications. Notably, LLMs often struggle with tasks involving token-level manipulation (Chen et al., 2024d), self-reflection (Xiong et al., 2024), and perceiving physical worlds (Fu et al., 2025), such as complex counting and verbalized confidence. These capabilities are believed to play an important role in developing trustworthy and explainable LLM-ITL solutions.*

4. Discussion: where next

While LLMs have demonstrated significant potential in "in-the-loop" solutions, persistent limitations hinder their effectiveness in specialized sub-tasks. For instance, they struggle with direct computational tasks such as optimization and quantitative trading (Zhao et al., 2024), where precise numerical reasoning is critical. Furthermore, studies suggest that single LLM agents may underperform human experts in forecasting accuracy (Schoenegger & Park, 2023) and exhibit reliability concerns due to inherent model variability and biases (Kholodna et al., 2024). These limitations raise questions about the consistency of generated outputs - such as rewards or feedback - in high-stakes applications (Cegin et al., 2023). Motivated by these challenges, we highlight key future research directions to advance LLM-in-the-loop frameworks and bridge gaps in reliability and adaptability.

Crowdsourcing with LLM. In human-in-the-loop applications, crowdsourcing is often employed to leverage the "wisdom of the crowd" in solving problems through collaborative efforts (Tong et al., 2019; Zhang et al., 2013; 2014). With the increasing use of ChatGPT by crowd workers on MTurk (Veselovsky et al., 2023), we argue that the emergence of LLM-driven crowds, such as "LMTurk" (Zhao et al., 2022), offers a promising foundation for developing more robust LLM-in-the-loop solutions and benefiting the implementation of the aforementioned techniques and applications. This approach harnesses diverse knowledge from different LLMs, helping to reduce biases and errors that might occur when relying on a single model (Kholodna et al., 2024). Recognizing the growing popularity of multi-agent LLM systems (Guo et al., 2024; Hong et al., 2024c), designing LLM crowdsourcing solutions from a multi-agent perspective is a promising research avenue (Jiang et al., 2018). Additionally, leveraging well-established theories in crowdsourcing, such as crowd selection, task decomposition, and result aggregation (Zhang et al., 2024a; Bhatti et al., 2020), provides a comprehensive framework to guide future research directions and technical advancements in LLM multi-agent systems and the "science of LLM-in-the-loop."

Text-to-Solution with LLM. Recent advancements in text-to-code generation (natural language to code) have demonstrated its efficacy in automating problem-solving through code synthesis, requiring minimal programming expertise (Guo et al., 2023; Nijkamp et al., 2023; Ni et al., 2023). However, designing effective LLM-in-the-loop solutions demands significant domain knowledge, such as creating optimal LLM utilities and integrated workflows. Automating this process via a novel "Text-to-Solution" framework could significantly enhance the accessibility and adoption of LLM-in-the-loop methodologies.

An interesting observation with the interpretable thinking process in DeepSeek-R1 (DeepSeek-AI, 2024; 2025) reveals that, under a zero-shot setting, the model is capable of: 1) capturing the concept of LLM integration and LLM-in-the-loop without explicit definition, 2) identifying suitable phases of LLM integration, and 3) deriving concrete implementation plans (see Appendix C). However, the generated code quality remains inconsistent, and there is a lack of sufficient understanding of in-the-loop techniques, which limits the diversity of solutions and still necessitates human experts to design the high-level framework. Inspired by the success of AutoML in automatically designing machine learning applications (Lindauer et al., 2024), further research is encouraged to explore **Automated In-the-loop (AutoITL)** as a promising "text-to-solution" framework to automate LLM utility selection and workflow construction, streamlining the creation of effective LLM-in-the-loop solutions.

5. Conclusion

The emergence of the "LLM-in-the-loop" paradigm marks a significant advancement in LLM application and machine learning research, offering a scalable and cost-effective alternative to human involvement while addressing critical challenges such as data scarcity and model interpretation. By categorizing methodologies into data, model, and task-centric, this paper established a systematic framework for designing future LLM-ITL applications, enhancing the adaptability and robustness of ML systems. With extensive discussion on application scenarios and technical advancements, such as LLM crowdsourcing, we encourage the research community to recognize the potential of "in-the-loop" methodologies and promote further research in better leveraging conventional machine learning algorithms in the era of LLMs.

Impact Statement

This position paper introduces a novel paradigm, LLM-in-the-loop, providing the first formal definition, various motivations, techniques, and application scenarios to support future advancements and exploration among researchers and industrial practitioners. In Section 3, we provided case studies and a comprehensive categorization of methodologies for integrating LLMs into machine learning development, emphasizing underexplored techniques and underutilized domains. Additionally, we discussed how LLMs can be further researched to support better in-the-loop solutions, encompassing multi-agent crowdsourcing and code generation. As the research community refines LLM-ITL methodologies, this paper lays the groundwork for leveraging the full potential of LLMs in tackling complex problems and inspiring new applications across various industries with the combined efforts of LLMs and machine learning.

References

Agarwal, E., Singh, J., Dani, V., Magazine, R., Ganu, T., and Nambi, A. Promptwizard: Task-aware prompt optimization framework, 2024. URL https://arxiv.org/abs/2405.18369.

Agarwal, N., Moehring, A., Rajpurkar, P., and Salz, T. Combining human expertise with artificial intelligence: Experimental evidence from radiology. Working Paper 31422, National Bureau of Economic Research, July 2023. URL http://www.nber.org/papers/w31422.

Ahn, J., Verma, R., Lou, R., Liu, D., Zhang, R., and Yin, W. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 225–237, 2024.

An, W., Shi, W., Tian, F., Lin, H., Wang, Q., Wu, Y., Cai, M., Wang, L., Chen, Y., Zhu, H., and Chen, P. Generalized category discovery with large language models in the loop. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8653–8665, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.512. URL https://aclanthology.org/2024.findings-acl.512/.

Arous, I., Dolamic, L., Yang, J., Bhardwaj, A., Cuccu, G., and Cudré-Mauroux, P. Marta: Leveraging human rationales for explainable text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):5868–5876, May 2021. doi: 10.1609/aaai.v35i7.16734. URL https://ojs.aaai.org/index.php/AAAI/article/view/16734.

Ba, Y., Mancenido, M. V., and Pan, R. Fill in the gaps: Model calibration and generalization with synthetic data. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17211–17225, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.955. URL https://aclanthology.org/2024.emnlp-main.955/.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Balek, V., Sỳkora, L., Sklenák, V., and Kliegr, T. Llm-based feature generation from text for interpretable machine learning. *arXiv preprint arXiv:2409.07132*, 2024.

Barj, H. N. E. and Sautory, T. Reinforcement learning from llm feedback to counteract goal misgeneralization. *arXiv preprint arXiv:2401.07181*, 2024.

Bartolo, M., Roberts, A., Welbl, J., Riedel, S., and Stenetorp, P. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8, 2020. doi: 10.1162/tacl\\_a\\_00338. URL https://aclanthology.org/2020.tacl-1.43.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, August 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.50. URL https://doi.org/10.1109/TPAMI.2013.50.

Bhattacharjee, A., Moraffah, R., Garland, J., and Liu, H. Towards llm-guided causal explainability for black-box text classifiers. In *AAAI 2024 Workshop on Responsible Language Models, Vancouver, BC, Canada*, 2024.

Bhatti, S. S., Gao, X., and Chen, G. General framework, opportunities and challenges for crowdsourcing techniques: A comprehensive survey. *Journal of Systems and Software*, 167:110611, 2020. ISSN 0164-1212. doi: https://doi.org/10.1016/j.jss.2020.110611. URL https://www.sciencedirect.com/science/article/pii/S0164121220300893.

Bi, Z., Zhang, N., Xue, Y., Ou, Y., Ji, D., Zheng, G., and Chen, H. OceanGPT: A large language model for ocean science tasks. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3357–3372, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.184. URL https://aclanthology.org/2024.acl-long.184/.

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

Cao, Y., Zhao, H., Cheng, Y., Shu, T., Chen, Y., Liu, G., Liang, G., Zhao, J., Yan, J., and Li, Y. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

Casanueva, I., Temčinas, T., Gerz, D., Henderson, M., and Vulić, I. Efficient intent detection with dual sentence encoders. In Wen, T.-H., Celikyilmaz, A., Yu, Z., Papangelis, A., Eric, M., Kumar, A., Casanueva, I., and Shah, R. (eds.), *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlp4convai-1.5. URL https://aclanthology.org/2020.nlp4convai-1.5/.

Cegin, J., Simko, J., and Brusilovsky, P. ChatGPT to replace crowdsourcing of paraphrases for intent classification: Higher diversity and comparable model robustness. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1889–1905, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.117. URL https://aclanthology.org/2023.emnlp-main.117.

Chang, S., Wang, R., Ren, P., and Huang, H. Enhanced short text modeling: Leveraging large language models for topic refinement. *arXiv preprint arXiv:2403.17706*, 2024.

Chen, B., Zhang, Z., Langrené, N., and Zhu, S. Unleashing the potential of prompt engineering in large language models: a comprehensive review, 2024a. URL https://arxiv.org/abs/2310.14735.

CHEN, C., Hu, Y., Yang, C.-H. H., Siniscalchi, S. M., Chen, P.-Y., and Chng, E. Hyporadise: An open baseline for generative speech recognition with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=cAjZ3tMye6.

Chen, L., Zaharia, M., and Zou, J. Frugalgpt: How to use large language models while reducing cost and improving performance, 2023. URL https://arxiv.org/abs/2305.05176.

Chen, L., Trivedi, A., and Velasquez, A. Llms as probabilistic minimally adequate teachers for dfa learning. *arXiv preprint arXiv:2408.02999*, 2024b.

Chen, R., Qin, C., Jiang, W., and Choi, D. Is a large language model a good annotator for event extraction? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17772–17780, 2024c.

Chen, X., Gao, C., Chen, C., Zhang, G., and Liu, Y. An empirical study on challenges for llm application developers. *ACM Transactions on Software Engineering and Methodology*, 2025a.

Chen, Y., Liu, Y., Yan, J., Bai, X., Zhong, M., Yang, Y., Yang, Z., Zhu, C., and Zhang, Y. See what LLMs cannot answer: A self-challenge framework for uncovering LLM weaknesses. In *First Conference on Language Modeling*, 2024d. URL https://openreview.net/forum?id=18iNTRPx8c.

Chen, Y., Ding, Z.-h., Wang, Z., Wang, Y., Zhang, L., and Liu, S. Asynchronous large language model enhanced planner for autonomous driving. In *European Conference on Computer Vision*, pp. 22–38. Springer, 2025b.

Choi, J., Yun, J., Jin, K., and Kim, Y. Multi-news+: Cost-efficient dataset cleansing via LLM-based data annotation. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15–29, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.2. URL https://aclanthology.org/2024.emnlp-main.2/.

Coden, A., Danilevsky, M., Gruhl, D., Kato, L., and Nagarajan, M. A method to accelerate human in the loop clustering. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 237–245. SIAM, 2017.

Dai, S.-C., Xiong, A., and Ku, L.-W. LLM-in-the-loop: Leveraging large language model for thematic analysis. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational*

Linguistics: EMNLP 2023, pp. 9993–10001, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.669. URL https://aclanthology.org/2023.findings-emnlp.669/.

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Diligenti, M., Roychowdhury, S., and Gori, M. Integrating prior knowledge into deep learning. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 920–923, 2017. doi: 10.1109/ICMLA.2017.00-37.

Ding, B., Min, Q., Ma, S., Li, Y., Yang, L., and Zhang, Y. A rationale-centric counterfactual data augmentation method for cross-document event coreference resolution. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1112–1140, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.63. URL https://aclanthology.org/2024.naacl-long.63/.

Du, Y., Watkins, O., Wang, Z., Colas, C., Darrell, T., Abbeel, P., Gupta, A., and Andreas, J. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*, pp. 8657–8677. PMLR, 2023.

Fang, Z., Alqazlan, L., Liu, D., He, Y., and Procter, R. A user-centered, interactive, human-in-the-loop topic modelling system. In Vlachos, A. and Augenstein, I. (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 505–522, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.37. URL https://aclanthology.org/2023.eacl-main.37/.

Fu, X., Hu, Y., Li, B., Feng, Y., Wang, H., Lin, X., Roth, D., Smith, N. A., Ma, W.-C., and Krishna, R. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pp. 148–166. Springer, 2025.

Gan, W., Wan, S., and Yu, P. S. Model-as-a-service (maas): A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pp. 4636–4645, 2023. doi: 10.1109/BigData59044.2023.10386351.

Geng, J., Cai, F., Wang, Y., Koeppl, H., Nakov, P., and Gurevych, I. A survey of confidence estimation and calibration in large language models. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6577–6595, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.366. URL https://aclanthology.org/2024.naacl-long.366/.

Gilardi, F., Alizadeh, M., and Kubli, M. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi: 10.1073/pnas.2305016120. URL https://www.pnas.org/doi/abs/10.1073/pnas.2305016120.

Grunde-McLaughlin, M., Lam, M. S., Krishna, R., Weld, D. S., and Heer, J. Designing llm chains by adapting techniques from crowdsourcing workflows, 2024. URL https://arxiv.org/abs/2312.11681.

Gu, Y., You, H., Cao, J., and Yu, M. Large language models for constructing and optimizing machine learning workflows: A survey. *arXiv preprint arXiv:2411.10478*, 2024.

Guo, D., Xu, C., Duan, N., Yin, J., and McAuley, J. Longcoder: a long-range pre-trained language model for code completion. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., and Zhang, X. Large language model based multi-agents: A survey of progress and challenges. In Larson, K. (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 8048–8057. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/890. URL https://doi.org/10.24963/ijcai.2024/890. Survey Track.

He, L., Michael, J., Lewis, M., and Zettlemoyer, L. Human-in-the-loop parsing. In Su, J., Duh, K., and Carreras, X. (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2337–2342, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1258. URL https://aclanthology.org/D16-1258/.

He, X., Lin, Z., Gong, Y., Jin, A.-L., Zhang, H., Lin, C., Jiao, J., Yiu, S. M., Duan, N., and Chen, W. AnnoLLM: Making large language models to be better crowdsourced annotators. In Yang, Y., Davani, A., Sil, A., and Kumar, A. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pp. 165–190, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-industry.15. URL https://aclanthology.org/2024.naacl-industry.15.

Hettiachchi, D., Sanderson, M., Goncalves, J., Hosio, S., Kazai, G., Lease, M., Schaekermann, M., and Yilmaz, E. Investigating and mitigating biases in crowdsourced data. In *Companion Publication of the 2021 Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '21 Companion, pp. 331–334, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384797. doi: 10.1145/3462204.3481729. URL https://doi.org/10.1145/3462204.3481729.

Hollmann, N., Müller, S., and Hutter, F. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. *Advances in Neural Information Processing Systems*, 36, 2024.

Holzinger, A. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain informatics*, 3(2):119–131, 2016.

Hong, M., Song, Y., Jiang, D., Ng, W., Sun, Y., and Zhang, C. J. Dial-in llm: Human-aligned dialogue intent clustering with llm-in-the-loop. *arXiv preprint arXiv:2412.09049*, 2024a.

Hong, M., Song, Y., Jiang, D., Wang, L., Guo, Z., and Zhang, C. J. Expanding chatbot knowledge in customer service: Context-aware similar question generation using large language models, 2024b. URL https://arxiv.org/abs/2410.12444.

Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z., Zhou, L., Ran, C., Xiao, L., Wu, C., and Schmidhuber, J. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024c. URL https://openreview.net/forum?id=VtmBAGCN7o.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Hu, Y., Chen, C., Yang, C.-H., Li, R., Zhang, D., Chen, Z., and Chng, E. GenTranslate: Large language models are generative multilingual speech and machine translators. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 74–90, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.5. URL https://aclanthology.org/2024.acl-long.5/.

Huang, X., Cheng, S., Shu, Y., Bao, Y., and Qu, Y. Question decomposition tree for answering complex questions over knowledge bases. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i11.26519. URL https://doi.org/10.1609/aaai.v37i11.26519.

Islam, T. and Goldwasser, D. Uncovering latent arguments in social media messaging by employing llms-in-the-loop strategy, 2024. URL https://arxiv.org/abs/2404.10259.

Jiang, J., An, B., Jiang, Y., Lin, D., Bu, Z., Cao, J., and Hao, Z. Understanding crowdsourcing systems from a multiagent perspective and approach. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 13(2):1–32, 2018.

Karimpanal, T. G., Semage, L. B., Rana, S., Le, H., Tran, T., Gupta, S., and Venkatesh, S. Lagr-seq: Language-guided reinforcement learning with sample-efficient querying. *arXiv preprint arXiv:2308.13542*, 2023.

Keles, B., Gunay, M., and Caglar, S. I. Llms-in-the-loop part-1: Expert small ai models for bio-medical text translation, 2024. URL https://arxiv.org/abs/2407.12126.

Kholodna, N., Julka, S., Khodadadi, M., Gumus, M. N., and Granitzer, M. Llms in the loop: Leveraging large language model annotations for active learning in low-resource languages. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 397–412. Springer, 2024.

Khot, T., Trivedi, H., Finlayson, M., Fu, Y., Richardson, K., Clark, P., and Sabharwal, A. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=_nGgzQjzaRy.

Kumar, V., Smith-Renner, A., Findlater, L., Seppi, K., and Boyd-Graber, J. Why didn't you listen to me? comparing user control of human-in-the-loop topic models.

In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6323–6330, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1637. URL https://aclanthology.org/P19-1637/.

Kuzman, T., Mozetič, I., and Ljubešić, N. Chatgpt: Beginning of an end of manual linguistic data annotation? use case of automatic genre identification, 2023. URL https://arxiv.org/abs/2303.03953.

Kwon, M. and Michael, S. Reward design with language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Laleh, A. R. and Ahmadabadi, M. N. A survey on enhancing reinforcement learning in complex environments: Insights from human and llm feedback. *arXiv preprint arXiv:2411.13410*, 2024.

Larson, S., Mahendran, A., Peper, J. J., Clarke, C., Lee, A., Hill, P., Kummerfeld, J. K., Leach, K., Laurenzano, M. A., Tang, L., and Mars, J. An evaluation dataset for intent classification and out-of-scope prediction. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1311–1316, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1131. URL https://aclanthology.org/D19-1131/.

Lebioda, K., Vorobev, V., Petrovic, N., Pan, F., Zolfaghari, V., and Knoll, A. Towards single-system illusion in software-defined vehicles–automated, ai-powered workflow. *arXiv preprint arXiv:2403.14460*, 2024.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703/.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020b. Curran Associates Inc. ISBN 9781713829546.

Liang, C., Du, H., Sun, Y., Niyato, D., Kang, J., Zhao, D., and Imran, M. A. Generative ai-driven semantic communication networks: Architecture, technologies and applications. *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2024. doi: 10.1109/TCCN.2024.3435524.

Lin, J., Diesendruck, M., Du, L., and Abraham, R. Batchprompt: Accomplish more with less. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Agyicd577r.

Lindauer, M., Karl, F., Klier, A., Moosbauer, J., Tornede, A., Mueller, A. C., Hutter, F., Feurer, M., and Bischl, B. Position: A call to action for a human-centered autoML paradigm. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=wELbEYgnmo.

Liu, W., Wang, X., Wu, M., Li, T., Lv, C., Ling, Z., JianHao, Z., Zhang, C., Zheng, X., and Huang, X. Aligning large language models with human preferences through representation engineering. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10619–10638, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.572. URL https://aclanthology.org/2024.acl-long.572/.

Liu, X., Eshghi, A., Swietojanski, P., and Rieser, V. Benchmarking natural language understanding services for building conversational agents. In *Increasing naturalness and flexibility in spoken dialogue interaction: 10th international workshop on spoken dialogue systems*, pp. 165–183. Springer, 2021.

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

Liu, Y., Gautam, S., Ma, J., and Lakkaraju, H. Confronting LLMs with traditional ML: Rethinking the fairness of large language models in tabular classifications. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3603–3620, Mexico City, Mexico, June 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.198. URL https://aclanthology.org/2024.naacl-long.198/.

Moradi, M., Moradi, M., and Guastella, D. C. Experience sharing and human-in-the-loop optimization for federated robot navigation recommendation. In *ICIAP Workshops (2)*, pp. 179–188, 2023. URL https://doi.org/10.1007/978-3-031-51026-7_16.

Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J., and Fernández-Leal, A. Human-in-the-loop machine learning: a state of the art. *Artif. Intell. Rev.*, 56(4):3005–3054, August 2022. ISSN 0269-2821. doi: 10.1007/s10462-022-10246-w. URL https://doi.org/10.1007/s10462-022-10246-w.

Movva, R., Balachandar, S., Peng, K., Agostini, G., Garg, N., and Pierson, E. Topics, authors, and institutions in large language model research: Trends from 17K arXiv papers. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1223–1243, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.67. URL https://aclanthology.org/2024.naacl-long.67.

Ni, A., Iyer, S., Radev, D., Stoyanov, V., Yih, W.-t., Wang, S. I., and Lin, X. V. Lever: learning to verify language-to-code generation with execution. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Nijkamp, E., Pang, B., Hayashi, H., Tu, L., Wang, H., Zhou, Y., Savarese, S., and Xiong, C. Codegen: An open large language model for code with multi-turn program synthesis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=iaYcJKpY2B_.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., McGuinness, L. A., Stewart, L. A., Thomas, J., Tricco, A. C., Welch, V. A., Whiting, P., and Moher, D. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372, 2021. doi: 10.1136/bmj.n71. URL https://www.bmj.com/content/372/bmj.n71.

Paige, A., Soubki, A., Murzaku, J., Rambow, O., and Brennan, S. E. Training LLMs to recognize hedges in dialogues about roadrunner cartoons. In Kawahara, T., Demberg, V., Ultes, S., Inoue, K., Mehri, S., Howcroft, D., and Komatani, K. (eds.), *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 204–215, Kyoto, Japan, September 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.sigdial-1.18. URL https://aclanthology.org/2024.sigdial-1.18/.

Pandey, R., Purohit, H., Castillo, C., and Shalin, V. L. Modeling and mitigating human annotation errors to design efficient stream processing systems with human-in-the-loop machine learning. *International Journal of Human-Computer Studies*, 160:102772, 2022. ISSN 1071-5819. doi: https://doi.org/10.1016/j.ijhcs.2022.102772. URL https://www.sciencedirect.com/science/article/pii/S1071581922000015.

Pang, A., Jang, H., and Fang, S. Generating descriptive explanations of machine learning models using llm. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 5369–5374. IEEE, 2024.

Pattnaik, A., George, C., Tripathi, R. K., Vutla, S., and Vepa, J. Improving hierarchical text clustering with LLM-guided multi-view cluster representation. In Dernoncourt, F., Preoţiuc-Pietro, D., and Shimorina, A. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 719–727, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.54. URL https://aclanthology.org/2024.emnlp-industry.54/.

Prakash, B., Oates, T., and Mohsenin, T. LLM augmented hierarchical agents. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023. URL https://openreview.net/forum?id=K5MfysX15Q.

Pu, X., Gao, M., and Wan, X. Summarization is (almost) dead, 2023. URL https://arxiv.org/abs/2309.09558.

Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. Data augmentation can improve robustness. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=kgVJBBThdSZ.

Rouzegar, H. and Makrehchi, M. Enhancing text classification through llm-driven active learning and human annotation. In *Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII)*, pp. 98–111, 2024.

Schoenegger, P. and Park, P. S. Large language model prediction capabilities: Evidence from a real-world forecasting tournament, 2023. URL https://arxiv.org/abs/2310.13014.

Smith, R., Fries, J. A., Hancock, B., and Bach, S. H. Language models in the loop: Incorporating prompting into weak supervision. *ACM/JMS Journal of Data Science*, 1 (2):1–30, 2024.

Song, Y.-F., He, Y.-Q., Zhao, X.-F., Gu, H.-L., Jiang, D., Yang, H.-J., and Fan, L.-X. A communication theory perspective on prompting engineering methods for large language models. *Journal of Computer Science and Technology*, 39(4):984–1004, 2024.

Sottana, A., Liang, B., Zou, K., and Yuan, Z. Evaluation metrics in the era of GPT-4: Reliably evaluating large language models on sequence to sequence tasks. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8776–8788, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.543. URL https://aclanthology.org/2023.emnlp-main.543/.

Spiegel, B. A., Yang, Z., Jurayj, W., Bachmann, B., Tellex, S., and Konidaris, G. Informing reinforcement learning agents by grounding language to markov decision processes. In *Workshop on Training Agents with Foundation Models at RLC 2024*, 2024.

Sreedhar, M. N., Rebedea, T., and Parisien, C. Unsupervised extraction of dialogue policies from conversations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19029–19045, 2024.

Srivastava, A., Zou, J., and Sutton, C. Clustering with a reject option: Interactive clustering as bayesian prior elicitation. In *33rd International Conference on Machine Learning: ICML 2016*, pp. 16–20, 2016.

Srivastava, S., Huang, C., Fan, W., and Yao, Z. Instances need more care: Rewriting prompts for instances with llms in the loop yields better zero-shot performance. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 6211–6232, 2024.

Srivatsa, K. A. and Kochmar, E. What makes math word problems challenging for LLMs? In Duh, K., Gomez, H., and Bethard, S. (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1138–1148, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.72. URL https://aclanthology.org/2024.findings-naacl.72/.

Sudhakar, A. V., Parthasarathi, P., Rajendran, J., and Chandar, S. Language model-in-the-loop: Data optimal approach to recommend actions in text games. In *ICML 2024 Workshop on Foundation Models in the Wild*, 2024. URL https://openreview.net/forum?id=Q8z6crH27o.

Sun, T., Shao, Y., Qian, H., Huang, X., and Qiu, X. Black-box tuning for language-model-as-a-service. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20841–20855. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/sun22e.html.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

Tan, Z., Li, D., Wang, S., Beigi, A., Jiang, B., Bhattacharjee, A., Karami, M., Li, J., Cheng, L., and Liu, H. Large language models for data annotation and synthesis: A survey. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 930–957, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.54. URL https://aclanthology.org/2024.emnlp-main.54/.

Tang, Y., Wang, Z., Qu, A., Yan, Y., Wu, Z., Zhuang, D., Kai, J., Hou, K., Guo, X., Zhao, J., Zhao, Z., and Ma, W. ItiNera: Integrating spatial optimization with large language models for open-domain urban itinerary planning. In Dernoncourt, F., Preoţiuc-Pietro, D., and Shimorina, A. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1413–1432, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.104. URL https://aclanthology.org/2024.emnlp-industry.104/.

Tong, Y., Zhou, Z., Zeng, Y., Chen, L., and Shahabi, C. Spatial crowdsourcing: a survey. *The VLDB Journal*, 29(1):217–250, August 2019. ISSN 1066-8888. doi: 10.1007/s00778-019-00568-7. URL https://doi.org/10.1007/s00778-019-00568-7.

Törnberg, P. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. *arXiv preprint arXiv:2304.06588*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Veselovsky, V., Ribeiro, M. H., and West, R. Artificial artificial artificial intelligence: Crowd workers widely use large language models for text production tasks, 2023. URL https://arxiv.org/abs/2306.07899.

Viswanathan, V., Gashteovski, K., Gashteovski, K., Lawrence, C., Wu, T., and Neubig, G. Large language models enable few-shot clustering. *Transactions of the Association for Computational Linguistics*, 12:321–333, 2024. doi: 10.1162/tacl\\_a\\_00648. URL https://aclanthology.org/2024.tacl-1.18.

Wang, Y., Zhong, W., Li, L., Mi, F., Zeng, X., Huang, W., Shang, L., Jiang, X., and Liu, Q. Aligning large language models with human: A survey, 2023. URL https://arxiv.org/abs/2307.12966.

Wei, V. J., Wang, W., Jiang, D., Song, Y., and Wang, L. Asr-ec benchmark: Evaluating large language models on chinese asr error correction, 2024. URL https://arxiv.org/abs/2412.03075.

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., and He, L. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381, 2022. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2022.05.014. URL https://www.sciencedirect.com/science/article/pii/S0167739X22001790.

Wu, Y., Tao, Y., Li, P., Shi, G., Sukhatmem, G. S., Kumar, V., and Zhou, L. Hierarchical llms in-the-loop optimization for real-time multi-robot target tracking under unknown hazards. *arXiv preprint arXiv:2409.12274*, 2024.

Xiong, M., Hu, Z., Lu, X., LI, Y., Fu, J., He, J., and Hooi, B. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Xu, C., Xu, Y., Wang, S., Liu, Y., Zhu, C., and McAuley, J. Small models are valuable plug-ins for large language models. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 283–294, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.18. URL https://aclanthology.org/2024.findings-acl.18/.

Yang, C., Chen, P., and Huang, Q. Can chatgpt's performance be improved on verb metaphor detection tasks? bootstrapping and combining tacit knowledge. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1016–1027, 2024a.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2), January 2019. ISSN 2157-6904. doi: 10.1145/3298981. URL https://doi.org/10.1145/3298981.

Yang, S., Sun, R., and Wan, X. A new benchmark and reverse validation method for passage-level hallucination detection. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=n20PghmZaD.

Yang, X., Zhao, H., Xu, W., Qi, Y., Lu, J., Phung, D., and Du, L. Neural topic modeling with large language models in the loop. *arXiv preprint arXiv:2411.08534*, 2024b.

Yu, Y., Zhuang, Y., Zhang, J., Meng, Y., Ratner, A. J., Krishna, R., Shen, J., and Zhang, C. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36, 2024.

Yuan, Q., Kazemi, M., Xu, X., Noble, I., Imbrasaite, V., and Ramachandran, D. Tasklama: probing the complex task understanding of language models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2025. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i17.29918. URL https://doi.org/10.1609/aaai.v38i17.29918.

Zeng, Z., Yu, J., Gao, T., Meng, Y., Goyal, T., and Chen, D. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tr0KidwPLc.

Zhang, C. J., Chen, L., Jagadish, H. V., and Cao, C. C. Reducing uncertainty of schema matching via crowdsourcing. *Proc. VLDB Endow.*, 6(9):757–768, July 2013. ISSN 2150-8097. doi: 10.14778/2536360.2536374. URL https://doi.org/10.14778/2536360.2536374.

Zhang, C. J., Tong, Y., and Chen, L. Where to: crowd-aided path selection. *Proc. VLDB Endow.*, 7(14):2005–2016, October 2014. ISSN 2150-8097. doi: 10.14778/2733085.

2733105. URL https://doi.org/10.14778/2733085.2733105.

Zhang, C. J., Liu, Y., Zeng, P., Wu, T., Chen, L., Hui, P., and Hao, F. Similarity-driven and task-driven models for diversity of opinion in crowdsourcing markets. *The VLDB Journal*, pp. 1–22, 2024a.

Zhang, H., Sediq, A. B., Afana, A., and Erol-Kantarci, M. Generative ai-in-the-loop: Integrating llms and gpts into the next generation networks, 2024b. URL https://arxiv.org/abs/2406.04276.

Zhang, R., Li, Y., Ma, Y., Zhou, M., and Zou, L. LLMaAA: Making large language models as active annotators. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 13088–13103, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.872. URL https://aclanthology.org/2023.findings-emnlp.872/.

Zhang, X., Zhang, J., Rekabdar, B., Zhou, Y., Wang, P., and Liu, K. Dynamic and adaptive feature generation with llm. *arXiv preprint arXiv:2406.03505*, 2024c.

Zhang, Y., Wang, Z., and Shang, J. ClusterLLM: Large language models as a guide for text clustering. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13903–13920, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.858. URL https://aclanthology.org/2023.emnlp-main.858/.

Zhao, H., Liu, Z., Wu, Z., Li, Y., Yang, T., Shu, P., Xu, S., Dai, H., Zhao, L., Jiang, H., Pan, Y., Chen, J., Zhou, Y., Mai, G., Liu, N., and Liu, T. Revolutionizing finance with llms: An overview of applications and insights, 2024. URL https://arxiv.org/abs/2401.11641.

Zhao, M., Mi, F., Wang, Y., Li, M., Jiang, X., Liu, Q., and Schuetze, H. LMTurk: Few-shot learners as crowdsourcing workers in a language-model-as-a-service framework. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V. (eds.), *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 675–692, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.51. URL https://aclanthology.org/2022.findings-naacl.51/.

Zhong, Z., Zhou, K., and Mottin, D. Harnessing large language models as post-hoc correctors. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14559–14574, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.867. URL https://aclanthology.org/2024.findings-acl.867/.

Zou, T., Liu, Y., Li, P., Zhang, J., Liu, J., and Zhang, Y.-Q. Fusegen: Plm fusion for data-generation based zero-shot learning. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2172–2190, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.130. URL https://aclanthology.org/2024.emnlp-main.130/.

Zytek, A., Pidò, S., and Veeramachaneni, K. Llms for xai: Future directions for explaining explanations, 2024. URL https://arxiv.org/abs/2405.06064.

A. Survey Methodology and Statistics

This paper primarily bases its supporting claims on systematic literature reviews. With reference to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) (Page et al., 2021), we outline the paper selection criteria in detail. Our review scope is strictly defined to include (1) the application of LLMs and (2) machine learning research. Research papers are sourced from a variety of channels, including peer-reviewed journals and conference proceedings. Our search strategy combines keyword searches with regulated filtering, focusing on publications from 2020 to 2024 to capture the latest advancements in LLM research. We prioritize papers from highly recognized peer-reviewed venues, specifically targeting top conferences and journals, such as ICML, NeurIPS, and ACL.

In recognition of the growing trend of disseminating emerging research through non-peer-reviewed preprints, we also collected studies submitted to e-print archive platforms such as arXiv. Our analysis of these preprints focused on extracting key insights, including novel definitions, design principles, optimization strategies, and newly proposed problems. Given the preliminary nature of these works, we emphasize their innovative ideas and concepts rather than their quantitative performance, acknowledging their lack of formal verification.

Category	Title	Year	Task
Task-Specific LLM-ITL	Neural Topic Modeling with **Large Language Models in the Loop** (Yang et al., 2024b)	2024	Topic Modeling
	LLMs as Probabilistic Minimally Adequate Teachers for DFA Learning (Chen et al., 2024b) *(...providing a theoretical foundation for automata learning with **LLMs in the loop**.)*	2024	DFA Learning
	Asynchronous Large Language Model Enhanced Planner for Autonomous Driving (Chen et al., 2025b) *(...we introduce AsyncDriver, a new asynchronous **LLM-enhanced closed-loop** framework)*	2024	Autonomous Driving
	Language Models in the Loop: Incorporating Prompting into Weak Supervision (Smith et al., 2024)	2022	Weak Supervision
	Dial-In LLM: Human-Aligned Dialogue Intent Clustering with **LLM-in-the-loop** (Hong et al., 2024a)	2024	Dialogue Clustering
	LLM-in-the-loop: Leveraging Large Language Model for Thematic Analysis (Dai et al., 2023)	2023	Thematic Analysis
Over-generalized ITL	Uncovering Latent Arguments in Social Media Messaging by Employing **LLMs-in-the-Loop Strategy** (Islam & Goldwasser, 2024)	2024	Social Media Analysis
	LLMs in the Loop: Leveraging Large Language Model Annotations for Active Learning in Low-Resource Languages (Kholodna et al., 2024)	2024	Active Learning
	Generalized Category Discovery with Large **Language Models in the Loop** (An et al., 2024)	2024	Category Discovery
	Generative AI-in-the-loop: Integrating LLMs and GPTs into the Next Generation Networks (Zhang et al., 2024b)	2024	Network Integration
Referential Works	Hierarchical **LLMs In-the-loop Optimization** for Real-time Multi-Robot Target Tracking under Unknown Hazards (Wu et al., 2024)	2024	Robotics
	Training LLMs to Recognize Hedges in Spontaneous Narratives (Paige et al., 2024) *(...we used an **LLM-in-the-Loop** approach to improve the gold standard coding)*	2024	Narrative Analysis
	LLMs-in-the-loop Part-1: Expert Small AI Models for Bio-Medical Text Translation (Keles et al., 2024)	2024	Bio-Medical Translation
	A Rationale-centric Counterfactual Data Augmentation Method for Cross-Document Event Coreference Resolution (Ding et al., 2024) *(...we develop a rationale-centric counterfactual data augmentation method with **LLM-in-the-loop**)*	2024	Coreference Resolution
	Towards Single-System Illusion in Software-Defined Vehicles – Automated, AI-Powered Workflow (Lebioda et al., 2024) *(...inclusion of modern generative AI, specifically **Large Language Models (LLMs), in the loop**)*	2024	Workflow Automation
	Instances Need More Care: Rewriting Prompts for Instances with **LLMs in the Loop** Yields Better Zero-Shot Performance (Srivastava et al., 2024)	2023	Zero-Shot Learning

Table 1. Existing works that explicitly mention "LLM-in-the-loop" in their titles or abstracts can be categorized as follows: "task-specific" includes studies that employed LLM-ITL for a single specific task, "over-generalized" encompasses works with a broad scope extending beyond LLMs, and "referential works" comprises publications that simply referenced the term without applying the methodology.

Venue	Year: Count	Total
Arxiv	2016:1, 2019:1, 2023: 9, 2024: 22, 2025:1	32
ACL	2019:1, 2020:3, 2023:1, 2024:12	17
EMNLP	2016:1, 2019:1, 2023:7, 2024:7	16
ICLR	2015: 1, 2022:1, 2023:3, 2024:4	9
NeurIPS	2014:1, 2017:1, 2020:2, 2012:1, 2023:1, 2024:2	8
ICML	2016:1, 2022:1, 2023:3, 2024:2	7
NAACL	2022:1, 2024:1	7
AAAI	2021:1, 2023:1, 2024:2, 2025:1	5
VLDB	<2020: 3, 2024:1	4
Other	<2022:10, 2022:3, 2023:4, 2024:9, 2025:3	29
Total		**134**

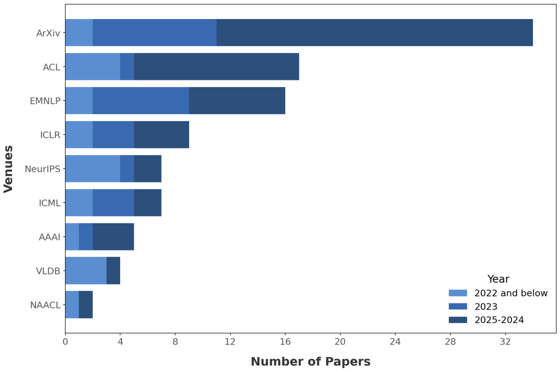


Figure 5. Summary of surveyed papers by publication venue and year: "Others" include TPAMI, PNAS, IJCAI, ECCV, BigData, etc., each with fewer than 2 papers included.

B. Empirical Study on LLM-Native Text Clustering

Experimental Setup. In this empirical study, the goal is to group n sentences into K clusters by directly prompting LLMs. Three widely adopted benchmark datasets are evaluated, namely CLINC150 (Larson et al., 2019), Banking77 (Casanueva et al., 2020), and HWU64 (Liu et al., 2021). The GPT-4o is employed via the OpenAI API for its broad accessibility, facilitating the reproducibility of results. To mitigate the inherent variability of LLMs while ensuring the significance of the findings, a "resampling" technique, as proposed in (Chen et al., 2024a), is implemented. The model is run 50 times with the same prompt and input data, with the temperature set to 0.5 to balance randomness and consistency in the outputs.

B.1. LLM-native Text Clustering with Prompt Engineering

An exploratory analysis shows that the LLM cannot handle the entire dataset due to input token constraints. Therefore, a subset of the dataset is sampled, consisting of 240 sentences divided into 8 clusters. The objectives of this experiment are twofold: 1) to assess the extent to which LLMs exhibit incapabilities under different prompts, as indicated by discrepancies in the generated solution space and the targeted space defined by the task requirement, and 2) to evaluate the clustering performance of usable LLM-generated cluster assignments. Three hand-crafted prompts were designed: a vanilla instruction prompt with the hint "each label corresponds to a sentence," based on the setup from (Kholodna et al., 2024); a few-shot prompt; and a chain-of-thought prompt. Additionally, the state-of-the-art prompt tuning method, PromptWizard (Agarwal et al., 2024), was used to generate two tailored prompts - one with reasoning steps and one without - specifically tuned to align solution space. Details of the tuning process and the experimented prompts are available in our GitHub repository.[2]

Based on the results presented in Table 2, it is evident that the LLM-naive approach underperform in the clustering task, with up to 98% of responses from the standard prompt and 90% from the best-performing prompt failing to align with the targeted label count, making these outputs largely ineffective and a waste of tokens. The adoption of more advanced prompting techniques shows a slight improvement, with prompt tuning without reasoning (i.e., "pw_wo_reasoning") providing the highest number of usable clustering results. While the expected generation of 240 labels remains problematic, the second requirement of clustering into 8 distinct clusters (i.e., adhering to the output space) is well met, with the best performing prompt successfully generating a list with exactly 8 labels without any error. However, the prompt tuning process incurs substantial costs, both during tuning and at inference time, where the instruction prompt becomes excessively lengthy, posing additional challenges. Additionally, a notable number of samples exceeded the targeted label count, contradicting the "laziness" or "output truncation" behavior of LLMs, which typically produce less when asked to generate more.

With the few correct samples obtained, the clustering performance was further evaluated against K-means, which achieved a perfect Normalized Mutual Information (NMI) score of 1. Analyzing the best-performing result from each prompting technique revealed that LLM-based clustering performs reasonably well for this simple task, with the top method achieving performance comparable to K-means clustering. The poorest performance was observed in the reasoning-based prompt, specifically tuned to instruction following, suggesting a potential trade-off between strictly following instructions to ensure usability of results and optimizing for task-solving performance. Despite this, concerns remain about the practicality of using LLMs for text clustering, as the number of usable results for this simple task is still significantly low, which raises doubts about their capability to manage increasing task complexity.

	CLINC150					**Banking77**					**HWU64**				
Prompt	L	E	G	OOS	NMI	L	E	G	OOS	NMI	L	E	G	OOS	NMI
vanilla	13	1	36	2	0.976	6	0	44	17	-	12	1	37	20	0.789
cot	19	1	30	1	0.909	13	2	35	15	0.763	7	0	43	12	-
fewshot	15	2	32	5	1	12	2	36	44	**0.858**	19	2	29	16	0.794
pw_wo_reasoning	15	**4**	31	**0**	**1**	6	**3**	41	25	0.760	0	2	48	17	**0.823**
pw_w_reasoning	14	2	34	3	0.896	5	0	45	24	-	6	0	44	27	-

Table 2. Summary of clustering results generated using various prompts, each repeated 50 times, under the clustering setting of $n = 240$ and $k = 8$. The statistics include counts of cases that are Less Than (L), Equal to (E), or Greater Than (G) the target number of clusters n; Out of Set (OOS) denotes misaligned label sets; and Normalized Mutual Information (NMI) measures the clustering quality for results with correct cluster counts and label sets, when applicable. The best results are highlighted in bold.

[2] The complete code and data are available at https://anonymous.4open.science/r/LLM-in-the-loop-4F42/.

B.2. Input Data and Task Complexity

The next step involves evaluating the impact of input data size and task complexity on the performance of LLM-natie solution. The input data size varies, ranging from 60 to 600 sentences, with the objective of examining both the emergence of output failure and the variance of the solution space, measured by the difference between the target label count and the predicted label count. The best-performing prompt identified in the previous discussion (i.e., pw_wo_reasoning) is utilized.

From the clustering results in Table 3, we show that a simpler task with $n = 60$ can be easily accomplished with only one error out of 50 runs. As task complexity increases, output failures increase dramatically and tend to appear in a random pattern when the number of sentences exceeds 120, corresponding to approximately 5200 input tokens plus 4300 tokens from the instruction prompt. Although this is far from the maximum input token limit, the lengthy input to the LLM poses significant challenges for instruction following during the inference process. By analyzing the variance of the generated clustering results, we observe from Figure 6 that as task complexity increases, the variance also grows. This results in more outliers, i.e., results that significantly deviate from the majority, leading to more uninterpretable behavior. **These observations explain why existing research rarely considers LLM-native baselines, largely due to the infeasibility and unusual behaviors of LLMs, necessitating further research to investigate the causes and potential solutions.**

Note that the discussed problem is significantly different from the Batch Prompt (Lin et al., 2024). In Batch Prompt, while the input to the LLM contains n instances and expects n outputs, the tasks being solved are independent and can be easily decomposed into individual prompts. For example, solving 10 math problems in a single prompt or across ten separate prompts. The main goal of Batch Prompt is to reduce the cost of repeated instructions. In contrast, for tasks like clustering and NER, the input must contain n instances and the solution space is strictly bonded by the input data.

Task Setting	L	E	G	OOS
$n = 60$	0	49	1	0
$n = 120$	0	1	49	0
$n = 180$	12	8	30	8
$n = 240$	15	4	31	0
$n = 300$	12	1	37	2
$n = 360$	10	3	37	3
$n = 420$	10	1	39	6
$n = 480$	16	0	34	10

Table 3. Summary of clustering results generated with different clustering setting.

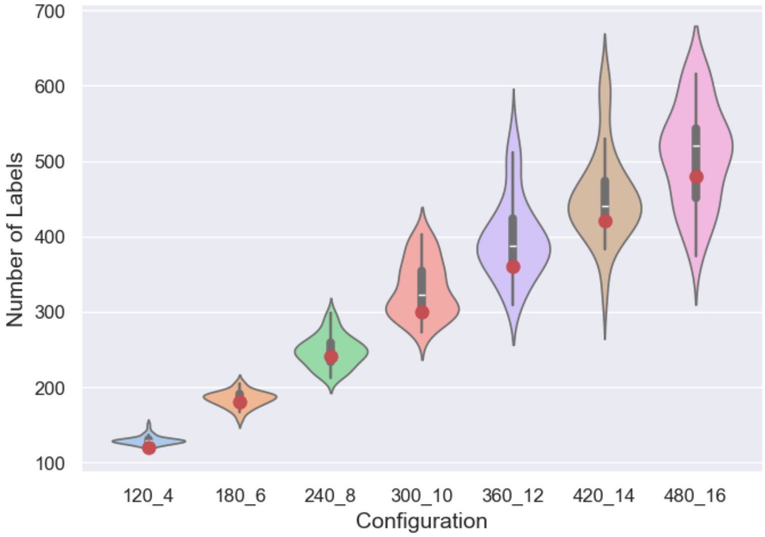
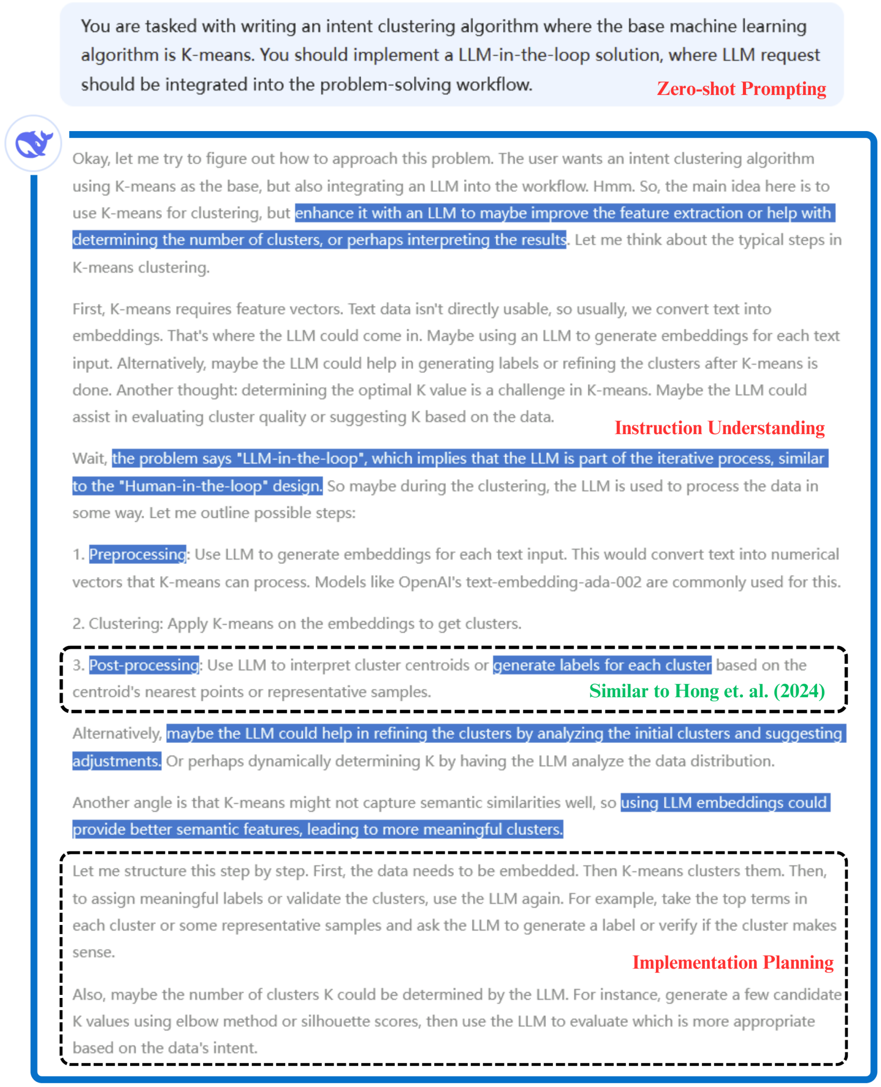


Figure 6. Variance of clustering results from the targeted solution space (i.e., for each specified number of clusters, n).

C. Demonstration: Text-to-Solution for Intent Clustering

In this demonstration, we aim to assess the practicality of generating LLM-in-the-loop solutions and evaluate whether the LLM can understand the concept of LLM-in-the-loop based on its existing knowledge. Two state-of-the-art models, DeepSeek-R1 and GPT-4o, are used under zero-shot settings and applied to solve the task of intent clustering. This task aligns with previously discussed case studies where various LLM-in-the-loop solutions have been implemented.



You are tasked with writing an intent clustering algorithm where the base machine learning algorithm is K-means. You should implement a LLM-in-the-loop solution, where LLM request should be integrated into the problem-solving workflow.

Okay, let me try to figure out how to approach this problem. The user wants an intent clustering algorithm using K-means as the base, but also integrating an LLM into the workflow. Hmm. So, the main idea here is to use K-means for clustering, but enhance it with an LLM to maybe improve the feature extraction or help with determining the number of clusters, or perhaps interpreting the results. Let me think about the typical steps in K-means clustering.

First, K-means requires feature vectors. Text data isn't directly usable, so usually, we convert text into embeddings. That's where the LLM could come in. Maybe using an LLM to generate embeddings for each text input. Alternatively, maybe the LLM could help in generating labels or refining the clusters after K-means is done. Another thought: determining the optimal K value is a challenge in K-means. Maybe the LLM could assist in evaluating cluster quality or suggesting K based on the data.

Wait, the problem says "LLM-in-the-loop", which implies that the LLM is part of the iterative process, similar to the "Human-in-the-loop" design. So maybe during the clustering, the LLM is used to process the data in some way. Let me outline possible steps:

1. Preprocessing: Use LLM to generate embeddings for each text input. This would convert text into numerical vectors that K-means can process. Models like OpenAI's text-embedding-ada-002 are commonly used for this.

2. Clustering: Apply K-means on the embeddings to get clusters.

3. Post-processing: Use LLM to interpret cluster centroids or generate labels for each cluster based on the centroid's nearest points or representative samples.

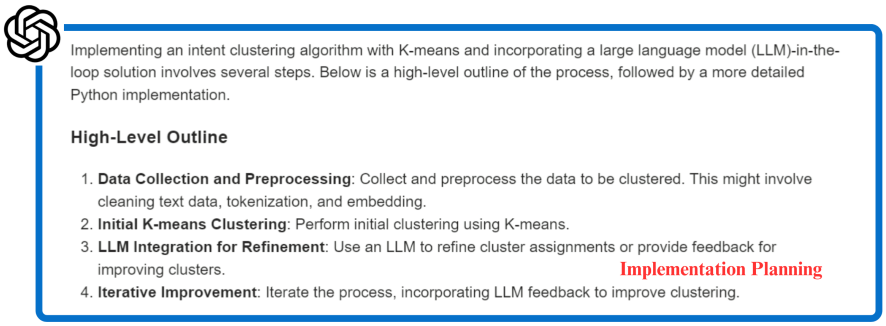
Alternatively, maybe the LLM could help in refining the clusters by analyzing the initial clusters and suggesting adjustments. Or perhaps dynamically determining K by having the LLM analyze the data distribution.

Another angle is that K-means might not capture semantic similarities well, so using LLM embeddings could provide better semantic features, leading to more meaningful clusters.

Let me structure this step by step. First, the data needs to be embedded. Then K-means clusters them. Then, to assign meaningful labels or validate the clusters, use the LLM again. For example, take the top terms in each cluster or some representative samples and ask the LLM to generate a label or verify if the cluster makes sense.

Also, maybe the number of clusters K could be determined by the LLM. For instance, generate a few candidate K values using elbow method or silhouette scores, then use the LLM to evaluate which is more appropriate based on the data's intent.

Figure 7. Text-to-solution with DeepSeek-R1 for LLM-in-the-loop Intent Clustering



Implementing an intent clustering algorithm with K-means and incorporating a large language model (LLM)-in-the-loop solution involves several steps. Below is a high-level outline of the process, followed by a more detailed Python implementation.

High-Level Outline

1. **Data Collection and Preprocessing**: Collect and preprocess the data to be clustered. This might involve cleaning text data, tokenization, and embedding.
2. **Initial K-means Clustering**: Perform initial clustering using K-means.
3. **LLM Integration for Refinement**: Use an LLM to refine cluster assignments or provide feedback for improving clusters.
4. **Iterative Improvement**: Iterate the process, incorporating LLM feedback to improve clustering.

Figure 8. Text-to-solution with GPT-4o for LLM-in-the-loop Intent Clustering