International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Reliability-Weighted Multi-Agent Annotation Workflow for Quality-Controlled LLM Labeling

Mr. Sarvagya Jain[1], Mr. Sandeep Piplotia[2], Ashish Shrivastava[3], Brajendra Prajapati[4]

[1,2]Assistant Professor, Department of CSE-AI/ML, Oriental Institute of Science and Technology , Bhopal

[3]Technical Assistance, Department of CSE-AI/ML, Oriental Institute of science and Technology, Bhopal

[4]Assistant Professor, Department of Information Technology, Barkatullah University

Abstract

Large Language Models (LLMs) are widely adopted for automatically labeling data because they work quickly and can handle large amounts of information. But there are problems like inconsistent results, made-up information, and different levels of reasoning ability across models, which can make the labeling less reliable. To fix these issues, this paper introduces a Multi-Agent Reliability-Weighted Annotation Workflow, a system designed to improve the trust in the labels created by LLMs.

The system uses 3 to 5 different types of LLM agents, each with different structures and ways of generating responses.

Each agent labels data on its own. The system then gives each agent a reliability score based on how well it has performed before, how confident it is in its answers, and how much the agents agree with each other. The final labels are created by combining the agents' results with more weight given to the more reliable ones, rather than just taking the most common answer. Cases where the model is unsure or there is a big disagreement are automatically marked for further checking by a person or for re-labeling.

Tests on datasets like AG News, CoNLL-2003, and SST-20 show that this method improves accuracy by up to 3.4% and increases label agreement by 0.06 compared to simpler methods that don't consider reliability.

Keywords: Large Language Models, Data Annotation, Multi-Agent Systems, Reliability Weighting, Quality Control, Weak Supervision, Automated Labeling

1. Introduction

There is a growing need for big datasets to support machine learning systems that use data. Traditionally, data labeling has been done by people or through online platforms where many people work together. However, this method is costly and takes a lot of time. Now, with the rise of large language models, data labeling can be done automatically on a large scale, offering a better and faster option for building data sets.

This paper presents the following contributions:

- We create a new process for labeling data using multiple agents, where each agent's reliability is taken into account. This helps in understanding how well each agent can do the task.

- We introduce a new way to measure reliability that uses past accuracy, confidence levels, and how much agreement there is between agents.
- We develop a system that checks for disagreements and highlights uncertain cases for re-checking by another agent or a person.
- We test this system on various tasks related to natural language processing and show that it performs better than other methods that use just one agent or do not consider reliability.

Even though using large language models for labeling has many advantages, there are some problems. These include making up incorrect labels, not being consistent in reasoning, and being too dependent on how the instructions are written. These issues can get worse when the same model is used again and again. To solve these problems, this paper suggests a new labeling workflow that uses multiple agents and focuses on reliability. This helps improve accuracy and control the quality of the labeled data.

2. Related Work:

Automated data labeling has been widely studied to cut down the cost and time needed to create big labeled datasets. Early work in this field looked at weak supervision, where many noisy labeling methods are used together to make probable labels. Systems like Snorkel use generative models to understand how accurate and related these labeling methods are, allowing for good label predictions without needing a lot of real labeled data [1].

Another main area of research is ensemble learning, where predictions from several classifiers or labelers are combined to make the results more reliable and better at generalizing.

Traditional methods in this area often use techniques like majority voting or simple averaging, which assume all labelers or models are equally good at their job [2], [3]. While these methods work in some cases, they don't take into account differences in how reliable each label or model is.With the rise of large language models, recent studies have looked at using them as automatic labels, either on their own or in groups.Some approaches use self-consistency, debating between models, or teamwork based on roles to improve the quality and reasoning of the labels [4]–[6]. Also, some systems involve humans in the process, where uncertain or disputed labels are sent to experts for review, helping to improve the accuracy of the final labels [7].Even with these improvements, most current methods that use multiple large language models treat all models as equally reliable, which isn't true.

In reality, these models can perform differently depending on the task, the subject matter, and the difficulty of the work. This assumption limits how well the methods can work, especially in tricky or specialized labeling tasks. The new method we're proposing takes into account how reliable each model is and uses a system that considers this reliability when combining results. It also uses disagreement to check and improve the quality of the labels, which fills an important gap in existing labeling systems.

2.1 Annotator Reliability Modeling

Classical methods for checking how reliable annotators are have been studied a lot in crowdsourcing research. Dawid and Skene created a probabilistic model to figure out the real labels and how accurate each annotator is, using a process called expectation-maximization. Later work like GLAD and Bayesian crowd labeling models built on this idea to take into account how hard a task is and how skilled the annotators are. Our approach is different because it doesn't assume that each annotator works independently or need complex steps to estimate hidden variables. Instead, we use built-in signals from large language models, such as their confidence and how much they agree with each other, to provide

reliable estimates in real time. This makes our method suitable for large-scale annotation systems that use modern large language models.

3. Problem Statement

Large Language Models help create scalable and cost-efficient automated data labels, but their predictions can be inconsistent, make up false information, and depend a lot on how prompts are designed. In single-agent systems, these issues lead to unreliable labels, especially for complicated or unclear data. Even though multi-agent systems reduce some of these problems through repetition, current methods like simple majority voting assume all agents are equally good and don't take into account differences in how reliable each agent is.

Also, current LLM-based labeling tools don't have a clear way to check how well each agent is doing, adjust their influence over time, or keep quality high with little help from humans.

This means unreliable agents can have a big impact on the final labels, causing lower quality results and more mistakes.

The main issue this work focuses on is the lack of a system that can handle different LLM agents while being reliable, scalable, and controlled in quality without needing much human help.

To solve this, we suggest a Multi-Agent Reliability-Weighted Annotation Workflow. This system estimates how reliable each agent is based on past performance, confidence levels, and how much the agents agree with each other. It also includes a way to escalate disagreements to ensure that the final labels are trustworthy.

4. Proposed MARW-AW Methodology

4.1 Notation

Let:

- $D = \{x_1, x_2, \dots, x_n\}$ denote the dataset of n instances
- $A = \{a_1, a_2, \dots, a_m\}$ denote the set of m LLM agents
- y_{ij} denote the label assigned by agent a_j to instance x_i
- Y denote the label space
- $C_{ij} \in [0,1]$ denote the confidence score reported by agent a_j for instance x_i
- w_j denote the reliability weight of agent a_j
- $\tau \in [0,1]$ denotes the disagreement threshold.

4.2 Methodology

The proposed workflow consists of four main stages, as illustrated below.

4.2.1 Multi-Agent Annotation

Several LLM agents work on independently annotating a data example with the same set of task instructions. It is done to capture diverse reasoning and prevent dependence on results from a solitary model.

Each agent $a_j \in A$ independently annotates each instance $x_i \in D$:

$$a_j(x_i) \rightarrow (y_{ij}, C_{ij}), \quad y_{ij} \in Y,\ C_{ij} \in [0,1]$$

This independence ensures diversity in reasoning and mitigates single-model bias.

4.2.2 Reliability Scoring

Each agent a_j is assigned a reliability score based on three components:

(5) Historical Accuracy

$$H_j \in [0,1]$$

computed on a held-out validation set.

(b) Mean Confidence

$$C^{-}_j = \frac{1}{n}\sum_{i=1}^{n} C_{ij}$$

I Inter-Agent Agreement

$$G_j \in [0,1]$$

To ensure temporal stability while allowing adaptability, each reliability component is updated using an exponential moving average (EMA):

$$H_j^{(t)} = \lambda H_j^{(t-1)} + (1-\lambda)\hat{H}_j^{(t)}$$
$$C^{-(t)}_j = \lambda C^{-(t-1)}_j + (1-\lambda)\hat{C}_j^{(t)}$$
$$G_j^{(t)} = \lambda G_j^{(t-1)} + (1-\lambda)\hat{G}_j^{(t)}$$

where $\hat{H}_j^{(t)}$, $\hat{C}_j^{(t)}$, and $\hat{G}_j^{(t)}$ denote the instantaneous estimates at iteration t, and $\lambda \in [0,1]$ controls the trade-off between stability and responsiveness.

While self-reported confidence estimates are not perfectly calibrated, we empirically find that relative confidence trends become informative when combined with historical accuracy and inter-agent agreement, mitigating noise from individual components and improving robustness.

4.2.3 Inter-Agent Agreement

Inter-agent agreement measures the consistency of each agent's annotations relative to other agents. For a given instance x_i, the agreement score for agent a_j is computed as:

$$\hat{G}_{ij} = \frac{1}{m-1}\sum_{k\neq j} I(y_{ij} = y_{ik})$$

where m is the total number of agents and $I(\cdot)$ is the indicator function.

The overall agreement score for agent a_j is then obtained by averaging across all annotated instances:

$$\hat{G}_j = \frac{1}{n}\sum_{i=1}^{n} \hat{G}_{ij}$$

This formulation ensures that agreement is computed in a pairwise, agent-centric manner, avoiding ambiguity in multi-annotator settings.

4.2.4 Reliability Weight Computation

To account for heterogeneous contributions of reliability components, we introduce weighting coefficients for historical accuracy, confidence, and inter-agent agreement."

The unnormalized reliability score for agent a_j is defined as:

$$\tilde{w}_j = \alpha H_j + \beta C^{-}_j + \gamma G_j$$

subject to:

$$\alpha + \beta + \gamma = 1, \quad \alpha, \beta, \gamma \geq 0$$

The normalized reliability weight is:

$$w_j = \frac{\widetilde{w}_j}{\sum_{k=1}^{m} \widetilde{w}_k}$$

subject to:

$$\sum_{j=1}^{m} w_j = 1, \quad w_j \geq 0$$

These parameters are then used to calculate a standardized reliability score for an agent.

4.2.4 Reliability-Weighted Aggregation

For classification tasks, the final aggregated label $\hat{y}_i$ for instance x_i is computed using weighted voting:

$$\hat{y}_i = \underset{y \in Y}{argmax} \sum_{j=1}^{m} w_j \cdot I(y_{ij} = y)$$

where $I(\cdot)$ is the indicator function:

$I(P) = \{1 \; if \; P \; is \; true, 0 \; otherwise \}$

4.2.5 Disagreement-Driven Quality Control

The disagreement score for instance x_i is defined as:

$$\delta_i = 1 - \sum_{j=1}^{m} w_j \cdot I(y_{ij} = \hat{y}_i)$$

An instance x_i is flagged if:

$$\delta_i > \tau$$

Let $\tau \in [0,1]$ denote a predefined disagreement threshold. Higher disagreement scores indicate weaker consensus among reliable agents, signaling instances that are more likely to benefit from re-annotation or expert review.

5. Proposed Algorithm (Pseudo Code)

```
# Multi-Agent Reliability-Weighted Annotation Workflow (MARW-AW)
defMARW_AW(D, A, tau, Y_set, alpha, beta, gamma, lambda_):
    n = len(D)
    m = len(A)
    # Initialize reliability components
    H = initialize_historical_accuracy(A)
    C_bar = [0.0for _ inrange(m)]
    G = [0.0for _ inrange(m)]
    w = [1.0 / m for _ inrange(m)]
    Y_hat = []
    Flags = []
    for I, x_i inenumerate(D):
        annotations = []
        confidences = []
        # Step 1: Independent annotation for j, agent inenumerate(A):
            y_ij, C_ij = agent.annotate(x_i)
            annotations.append(y_ij)
            confidences.append(C_ij)
```

```
    # Step 2: Instantaneous confidence update for j inrange(m):
        C_bar[j] = lambda_ * C_bar[j] + (1 – lambda_) * confidences[j]
    # Step 3: Instantaneous agreement update for j inrange(m):
        agreement = 0.0for k inrange(m):
            if j != k and annotations[j] == annotations[k]:
                agreement += 1
        agreement /= (m – 1)
        G[j] = lambda_ * G[j] + (1 – lambda_) * agreement
    # Step 4: Reliability weight computation
    w_tilde = [
        alpha * H[j] + beta * C_bar[j] + gamma * G[j]
        for j inrange(m)
    ]
    weight_sum = sum(w_tilde)
    w = [w_j / weight_sum for w_j in w_tilde]
    # Step 5: Weighted aggregation
    label_scores = {y: 0.0for y in Y_set}
    for j inrange(m):
        label_scores[annotations[j]] += w[j]

    y_hat_i = max(label_scores, key=label_scores.get)
    Y_hat.append(y_hat_i)
    # Step 6: Disagreement computation
    agreement_weight = sum(
        w[j] for j inrange(m) if annotations[j] == y_hat_i
    )
    delta_i = 1.0 – agreement_weight
    if delta_i > tau:
        Flags.append(x_i)
return Y_hat, Flags
```

6. Experimental Setup & Results

The proposed workflow was evaluated on multiple annotation tasks across diverse datasets. Performance was measured using annotation accuracy, inter-annotator agreement, and error rate metrics.

Results indicate that the reliability-weighted approach consistently outperforms single-agent annotation and unweighted multi-agent baselines. The workflow also demonstrates improved robustness in ambiguous cases and reduces the need for extensive human intervention.

6.1 . Dataset Details

Table 6.1.1 summarizes the datasets used for evaluating the annotation workflows across the three selected NLP tasks. All datasets contain expert-annotated ground truth labels, which serve as the reference standard for evaluation.

Table 6.1.1 : Dataset Details

Task	Dataset	#Instances	#Classes / Tags	Domain
Text Classification	AG News	120,000	4 classes	News
Named Entity Recognition	CoNLL-2003	14,041	4 entity types	General
Sentiment Analysis	SST-2	67,349	2 classes	Reviews

6.2. Quantitative Results

The proposed reliability-weighted multi-agent workflow was compared against two baseline methods: single-agent LLM annotation and unweighted multi-agent majority voting.

All reported metrics are averaged over three independent runs, with standard deviations below 0.7%.

1) Overall Annotation Performance

Table 6.2.1 : Annotation Performance Comparison

Method	Accuracy	F1-score	Cohen's κ
Single-Agent LLM	81.4	80.6	0.72
Majority Voting (Unweighted)	85.9	85.1	0.78
Proposed Reliability-Weighted	**89.3**	**88.7**	**0.84**

The results demonstrate that multi-agent aggregation significantly outperforms single-agent annotation. Moreover, incorporating reliability weighting yields further gains across all metrics, particularly in inter-annotator agreement (κ).

2) Task-wise Performance Breakdown

Table 6.2.2 Task-wise Accuracy (%)

Method	Text Classification	NER	Sentiment Analysis
Single-Agent LLM	83.2	79.1	82.0
Majority Voting	87.0	83.4	87.3
Proposed Method	**90.8**	**87.6**	**89.4**

The proposed workflow consistently achieves the highest accuracy across all tasks, with the largest improvements observed in NER, where annotation complexity and ambiguity are higher.

6.3. Disagreement-Driven Quality Control Analysis

To assess the effectiveness of disagreement-based filtering, instances with disagreement scores exceeding the threshold τ were flagged for re-annotation or human review.

All reported metrics are averaged over three independent runs, with standard deviations below 0.7%.

Table 6.3.1: Impact of Disagreement Filtering

Method	Flagged Samples (%)	Accuracy After Review
Majority Voting	18.6	87.1
Proposed Method	**12.4**	**91.2**

The reliability-weighted approach reduces the number of flagged samples while achieving higher post-review accuracy, indicating improved confidence estimation.

6.4. Statistical Significance Analysis

To verify whether the observed improvements are statistically significant, **paired t-tests** were conducted between the proposed method and baseline approaches across all tasks.

- Null hypothesis (H_0): No significant difference between methods
- Significance level: $\alpha = 0.05$

Table 6.4.1 : Statistical Significance Results (p-values)

Comparison	Accuracy	F1-score	Cohen's κ
Proposed vs Single-Agent	< 0.001	< 0.001	< 0.001
Proposed vs Majority Voting	0.003	0.005	0.002

All p-values fall well below the chosen significance threshold, confirming that the improvements introduced by the proposed reliability-weighted workflow are **statistically significant**.

6.5 Ablation Study on Reliability Components

We evaluate multiple model configurations to assess the contribution of each component. Specifically, we test:

Full model (H + C + G): the complete system incorporating all components.

Only historical accuracy: a simplified variant that relies solely on historical performance.

H + C: a model excluding the G component to measure its impact.

H + G: a model excluding the C component to measure its impact.

C + G: a model excluding the H component to measure its impact.

Unweighted majority voting (baseline): a baseline approach where all components contribute equally without learned weights.

This evaluation framework allows us to quantify the individual and combined effects of each component relative to a simple baseline.

Table 6.5.1: Ablation Study Results

Method	Accuracy	F1	κ
Full (H+C+G)	**89.3**	**88.7**	**0.84**
H only	**86.8**	**86.1**	**0.79**

H + C	**88.1**	**87.4**	**0.81**
H + G	**88.7**	**88.0**	**0.82**
C + G	**85.9**	**85.1**	**0.78**

"Historical accuracy contributes most to reliability estimation, while confidence and agreement provide complementary gains, particularly in ambiguous instances."

6.6 Implementation Details

We employ 3–5 heterogeneous LLM agents per task, instantiated using different model architectures and decoding temperatures. All agents receive identical task instructions to isolate reasoning variability. Confidence scores are elicited by prompting agents to self-report certainty on a normalized scale. Reliability coefficients are tuned on a held-out validation set comprising 10% of the data.

7. Discussion

The experimental results validate the central hypothesis that treating LLM agents as equally reliable annotators is suboptimal. By explicitly modeling agent reliability and incorporating disagreement-driven quality control, the proposed workflow produces annotations that are both more accurate and more consistent with expert labels. These benefits are especially pronounced in complex sequence-labeling tasks such as NER.

The proposed framework may be less effective when all agents share similar inductive biases or when confidence estimates are systematically miscalibrated. Additionally, maintaining multiple agents introduces computational overhead, which must be balanced against annotation quality gains in large-scale deployments.

7.1 Computational Complexity:

For each instance, the annotation cost scales linearly with the number of agents $O(m)$, while agreement computation incurs $O(m^2)$ overhead. In practice, with $m \leq 5$, this cost remains negligible relative to LLM inference time.

8. Conclusion and Future Work

8.1. Conclusion

This paper presented a **multi-agent reliability-weighted annotation workflow** designed to improve the quality and trustworthiness of annotations generated by large language models. Unlike existing approaches that treat all annotating agents equally, the proposed method explicitly models **agent reliability** and integrates a **disagreement-driven quality control mechanism**. By assigning adaptive weights to agents during annotation aggregation, the workflow effectively mitigates the impact of unreliable or inconsistent agents.

Extensive experiments conducted on text classification, named entity recognition, and sentiment analysis tasks demonstrate that the proposed approach consistently outperforms single-agent annotation and unweighted multi-agent majority voting. Improvements were observed across accuracy, F1-score, and Cohen's Kappa, with statistical significance, confirming both higher annotation correctness and stronger agreement with expert-labeled ground truth. The results highlight the importance of reliability-aware aggregation in multi-agent LLM systems, particularly for complex and ambiguous annotation tasks.

8.2. Future Work

Several exciting areas for future research come from this study. One idea is to improve the reliability estimation method by adding task-specific and domain-adapted models. This would let agents change their influence based on different types of data. Another direction is to use online learning techniques to update reliability in real time as data changes.

Future work might also look into human-in-the-loop methods that include human feedback to help adjust agent reliability and set better disagreement limits.

Expanding the framework to handle tasks like image-text or audio-text labeling is also a logical next step. Lastly, studying how well the process works with unreliable or poor-quality agents could make it more useful in real-world settings.

9. Ethics & Responsible Use

While the proposed workflow improves annotation reliability, automated labeling systems should not be treated as substitutes for expert judgment in high-stakes domains. Disagreement-based escalation mechanisms play a critical role in preventing silent error propagation and supporting responsible dataset construction.

10. References

1. A. Ratner et al., “Snorkel: Rapid Training Data Creation with Weak Supervision,” Proc. VLDB Endowment, vol. 11, no. 3, pp. 269–282, 2017.
2. T. Dietterich, “Ensemble Methods in Machine Learning,” Multiple Classifier Systems, Springer, pp. 1–15, 2000.
3. L. Rokach, “Ensemble-based classifiers,” Artificial Intelligence Review, vol. 33, no. 1–2, pp. 1–39, 2010.
4. X. Wang et al., “Self-Consistency Improves Chain-of-Thought Reasoning in Language Models,” Proc. ICLR, 2023.
5. L. Du et al., “Improving Factuality and Reasoning in LLMs through Multi-Agent Debate,” arXiv preprint arXiv:2305.19118, 2023.
6. S. Li et al., “CAMEL: Communicative Agents for ‘Mind’ Exploration of Large Language Models,” arXiv preprint arXiv:2303.17760, 2023.
7. A. Holzinger, “Human-in-the-Loop Machine Learning,” KI – Künstliche Intelligenz, vol. 30, pp. 105–111, 2016.