# Efficient Data Labeling by Hierarchical Crowdsourcing with Large Language Models

**Haodi Zhang[1,4], Junyu Yang[1], Jinyin Nie[1], Peirou Liang[1], Kaishun Wu[4], Defu Lian[3], Rui Mao[1], Yuanfeng Song[2]***

[1]Shenzhen University, [2]WeBank Co., Ltd, [3]University of Science and Technology of China
[4]The Hong Kong University of Science and Technology (Guangzhou)

## Abstract

Large language models (LLMs) have received lots of attention for their impressive performance in in-context dialogues and their potential to revolutionize service industries with a new business model, Model-as-a-Service (MaaS). Automated data labeling is a natural and promising service. However, labeling data with LLMs faces two main challenges: 1) the labels from LLMs may contain uncertainty, and 2) using LLMs for data labeling tasks can be prohibitively expensive, as the scales of datasets are usually tremendous. In this paper, we propose a hierarchical framework named **LMCrowd** that leverages multiple LLMs for efficient data labeling under budget constraints. The proposed LMCrowd framework first aggregates labels from multiple freely available LLMs, and then employs a large, paid MaaS LLM for relabeling selected instances. Furthermore, we formalize the core process as an optimization problem, aiming to select the optimal set of instances for relabeling by the MaaS LLM, given the current belief state. Extensive experimental evaluations across various real-world datasets demonstrate that our framework outperforms human labelers and GPT-4 in terms of both accuracy and efficiency.

## 1 Introduction

As machine learning models continue to grow in size and complexity, high-quality data has become an increasingly important factor for achieving good performances. However, as the scale of these models continue to grow at a rapid pace, the availability of high-quality training data could become a potential bottleneck (Lipton, 2018). This has led to a growing need for alternative approaches to data labeling that can keep pace with the demands of modern machine learning. One commonly-used method for labeling data is through crowd-based annotation (Li et al., 2016), which involves outsourcing the task to a large group of human annotators. However, this approach faces several challenges, including a shortage of qualified annotators (Sheng et al., 2008), the potential for errors and biases (Chapman et al., 2020; Kotamraju and Blanco, 2021; Zhang et al., 2023), and the high costs of managing a large workforce (Mason and Watts, 2009). These limitations have spurred interest in automated approaches to data labeling, which can leverage the power of machine learning methods and tools to provide faster, more accurate results.

With the recent advancements in large language models (LLMs, e.g., GPT-4 (OpenAI, 2023)), automated data labeling using LLMs has emerged as a promising alternative to human annotators. These LLMs have shown remarkable performance in various natural language processing (NLP) tasks, including question answering, sentiment analysis, and text classification (Brown et al., 2020). Automated data labeling with LLMs not only saves time and reduces cost but also ensures scalability in large datasets. Despite the advantages of leveraging LLMs to data labeling, two main challenges arise. Firstly, the labels generated by LLMs may contain noise, commonly referred to as hallucination, due to inherent errors in the models' predictions (Karimi et al., 2021). This could adversely impact the quality of the labeled data, potentially leading to incorrect training of downstream NLP models. Secondly, although using LLMs for data labeling costs less than hiring human labelers, it can still be significant, particularly for larger datasets. These constraints hinder the feasibility of automated labeling with LLMs, especially for smaller organizations or research teams with tight budgets.

To address these challenges, we propose **LMCrowd**, a novel framework that efficiently utilizes multiple LLMs for automated data labeling under limited budget constraints. LMCrowd is designed by the following three observations. First, open-

*Yuanfeng Song is the corresponding author.

*Proceedings of the 31st International Conference on Computational Linguistics*, pages 11290–11303
January 19–24, 2025. ©2025 Association for Computational Linguistics

source LLMs that can be easily deployed on local devices, while free of charge, are often not sufficiently powerful or accurate to accomplish high-quality data annotation tasks compared to their larger, paid MaaS counterparts like GPT-4. Secondly, while paid MaaS LLMs offer superior performance, their utilization for labeling entire datasets can be prohibitively expensive. For instance, the usage rate of ChatGPT (Radford et al., 2019a) service is approximately $\$0.06$ per $1$K tokens. For a dataset of the scale similar with QQP (one of the datasets in GLUE benchmark (Wang et al., 2018)), it would cost around $\$9600$ to label all the data instances using this service. If the problem is more difficult and requires a longer prompt, the costs would further escalate. Lastly, real-world labeling queries are often correlated with one another. By leveraging the correlations among labeling tasks to avoid sending every single task to LLMs for annotation, LMCrowd accomplishes efficient automated data annotation while minimizing the number of calls to costly MaaS LLMs. Specifically, the LMCrowd framework first utilizes multiple freely available LLMs to initialize labeling for unlabeled data and employs classical crowdsourcing aggregation algorithms to aggregate the initial label distributions. Subsequently, a selection-collection-update process is conducted, utilizing MaaS LLMs to provide feedback and update labels iteratively until the budget constraints are met. We formulate the selection of the set of tasks as an optimization problem, prove its NP-hardness, and provide an approximate algorithm that efficiently solves the problem.

To sum up, the main contribution of this paper is threefold.

- We propose a hierarchical framework that harnesses the capabilities of multiple LLMs to efficiently label data under limited budget constraints. This approach synergistically combines the advantages of freely available LLMs, which can independently annotate the entire dataset, and paid MaaS LLMs, which can selectively relabel data instances while operating within a specified budget constraint.

- We establish the core optimization challenge of identifying an appropriate set of tasks for the MaaS LLMs given the current information entropy distribution. We prove that the computation of the exact optimal solution is NP-hard, and subsequently propose an approximate algorithm that solves the problem efficiently.

- We perform comprehensive experiment to confirm that LMCrowd framework achieves high labelling quality with up to $80\%$ cost savings compared to relying solely on MaaS labeling, and exponential cost savings compared to manual labeling efforts.

## 2 Preliminary: Data Model

We approach a general dataset as a set of discrete multiple-choice tasks, where each task presents a question of the form "which label/labels should be annotated to the data instance?", given a label space $L$. This multiple-choice task can always be decomposed into several correlated binary queries, such as "should the instance be labeled as $l$?" for each label $l$ in $L$. For the rest of this paper, we refer to such a binary labeling task simply as a task and a dataset as a set of tasks, denoted by $T$. For instance, in a sentiment analysis dataset, a task $t \in T$ would present the sentence *'She loves this story'*, which can be labeled as true or false. Free LLMs can utilize the task directly as input for labeling, while MaaS LLMs may require further relabeling.

Similar to existing related works (Cheng et al., 2008; Lai et al., 2021; Zhang et al., 2018, 2023), we address the problem of analyzing $n$ labeling queries by treating each task as a Bernoulli random variable. The correlations among labeling queries are captured precisely by their joint distribution, covering $2^n$ possible truth-value interpretations. In a deterministic world, these interpretations are mutually exclusive, each representing a potential data state. Throughout the remainder of this paper, we refer to these interpretations as *profiles* of labels. The probability of a specific profile $\mathbf{r}$, denoted by $P(\mathbf{r})$, indicates the likelihood that $\mathbf{r}$ represents the true deterministic state of the labels. The set of all profiles is denoted by $\mathcal{R}$. If a given task $t$ is considered true in a particular profile $\mathbf{r}$, then we refer to $\mathbf{r}$ as a model of $t$, which is denoted as $\mathbf{r} \supset t$. Since each profile $\mathbf{r}$ in a labeling task set $T$ is an interpretation that covers all labeling queries in $T$, we can equivalent express $\mathbf{r} \not\supset t$ as $\mathbf{r} \supset \neg t$ for any task $t \in T$. It is noteworthy that different profiles are mutually exclusive, and the labeling queries in $T$ may not be independent.

Intuitively, a distribution on the profile space $\mathcal{R}$ represents a belief state of the data. Our approach seeks to leverage model-sourced response to update the belief state and enhance data quality. In our method, named LMCrowd, we use *Shannon's Information Entropy*, the same mathematical metric as in many existing works (Cheng et al., 2008; Lai et al., 2021; Zhang et al., 2018, 2023), for assessing the quality of query sets.

**Definition 1 (labeled data quality)** *Given a set of labeling task $T$ and its corresponding profile space $\mathcal{R}$, the evaluation of quality of $T$, represented by $Q(T)$ as a utility function, is formulated by the negative value of Shannon Entropy,*

$$\begin{aligned} P(\mathbf{r}) &= \prod_{f_i \in \mathbf{r}} P(f_i) \\ Q(T) &= -H(\mathcal{R}) = \sum_{\mathbf{r} \in \mathcal{R}} P(\mathbf{r}) \log P(\mathbf{r}) \end{aligned} \tag{1}$$

*where $\sum_{\mathbf{r} \in \mathcal{R}} P(\mathbf{r}) = 1$ and $f_i$ represents the case where the i-th data in $\mathbf{r}$ is ground-truth.*

The $Q$ value presented above serves as a utility function for evaluating the quality of a model-sourced result set.

The information entropy mentioned above serves as a measure of uncertainty of random variables. When dealing with a set of unlabeled data instances, the information entropy tends to be high due to the uniform distribution of labels, unless additional information is provided. The aim of our proposal is to effectively reduce this uncertainty, in other words, to improve the quality of the dataset, by leveraging various types of language models.

## 3 Our Method: LMCrowd

This segment provides a comprehensive presentation of our proposed framework, LMCrowd, which comprises three fundamental components: label aggregator, relabeling task selector and label distribution updater, as illustrated in Figure 1.

### 3.1 LLMs as a Crowd for Data Annotation

### 3.1.1 Labeling by LLMs

Labeling tasks are delegated to LLMs and answers collected are utilized to update the distribution and enhence the quality of data. Consider a labeling task set $T = \{t_1, t_2, \dots, t_n\}$ and a subset $T' \subseteq T$ that is dispatched to a particular LLM $m$ for labeling. The crowdsourced results for $T'$ are represented by $A(T') = \{a(t) | t \in T'\}$, where $a(t)$ can be either $true$ or $false$ , respectively. For a task $t \in T \backslash T'$, no results are obtained from $m$, as it is not assigned to label or relabel. In the following, we call such a subset $T'$ sent to LLMs a *query set*. The result set $A(T')$ is actually a partial profile of the labeling tasks, consisting of the assignments for the labeling tasks in $T'$. Therefore, we can extend the modeling relation and denote $A(T') \supset t$ if the task $t$ is interpreted as true in the collected answer $A(T')$. However, unlike a profile, a result set does not constitute a comprehensive mapping across $T$.. Thus, $A(T') \not\supset t$ does not imply $A(T') \supset \neg t$. For instance, consider the following query set and result set,

$$\begin{aligned} T &= \{t_1, \dots, t_n\} \\ T' &= \{t_1, t_2\} \\ A(T') &= \{a(t_1) = true, a(t_2) = false\} \end{aligned} \tag{2}$$

Obviously, $A(T') \not\supset t_3$, but $A(T') \not\supset \neg t_3$, since $t_3$ can not be derived from the result set $A(T')$. For a result answer $A(T')$ and a profile $\mathbf{r}$, their assignments for those tasks in $T'$ could be different.

**Definition 2 (consistent set and inconsistent set)** *For a profile* $\mathbf{r}$ *and a result answer $A(T')$, their* consistent *and* inconsistent *sets are respectively,*

$$\begin{aligned} cons(\mathbf{r}, A(T')) &= \{t \mid t \in T' \wedge (\mathbf{r} \supset t \Leftrightarrow A(T') \supset t)\} \\ incons(\mathbf{r}, A(T')) &= \{t \mid t \in T' \wedge (\mathbf{r} \supset t \Leftrightarrow A(T') \supset \neg t)\} \end{aligned} \tag{3}$$

Consider a query set $T'$ comprising of $k$ labeling queries, there are $|\{true, false\}^{T'}| = 2^k$ different result sets, and we denote the space of all possible result sets as $\mathcal{A}(T')$.

**Definition 3 (result set entropy)** *Given a set of tasks $T$ and a query set $T' \subseteq T$, we define the entropy of the resulting sets $H(\mathcal{A}(\mathcal{T}'))$ is*

$$H(\mathcal{A}(T')) = - \sum_{A(T') \in \mathcal{A}(T')} P(A(T')) \log P(A(T')) \tag{4}$$

It is worth noting that for the labeling queries that are not in $T'$, the answer set $A(T')$ fails to provide any information about them. Therefore, their assignments are not included in either $cons(\mathbf{r}, A(T'))$ or $incons(\mathbf{r}, A(T'))$. For each query $t$ in $T'$, the collected label $a_t$ can be correct or incorrect. The probability that $a_t$ aligns with the ground truth is equivalent with the accuracy rate of the LLM $m_t$ who completes the labeling task $t$, denoted as $Pr_{m_t}$. Conversely, the probability that $a_t$ is a wrong answer is $1 - Pr_{m_t}$. The accuracy rate $Pr_{m_t}$ of model $m_t$ can be estimated by a validation set of the data.

$$\begin{aligned} P(A(T')) = \sum_{\mathbf{r} \in \mathcal{R}} (P(\mathbf{r}) \cdot \prod_{t \in cons(\mathbf{r}, A(T'))} & Pr_{m_t} \cdot \\ \prod_{t \in incons(\mathbf{r}, A(T'))} & (1 - Pr_{m_t})) \end{aligned} \tag{5}$$

As aforementioned, after dispatching the chosen query set $T'$, the outcome set for $T'$ becomes

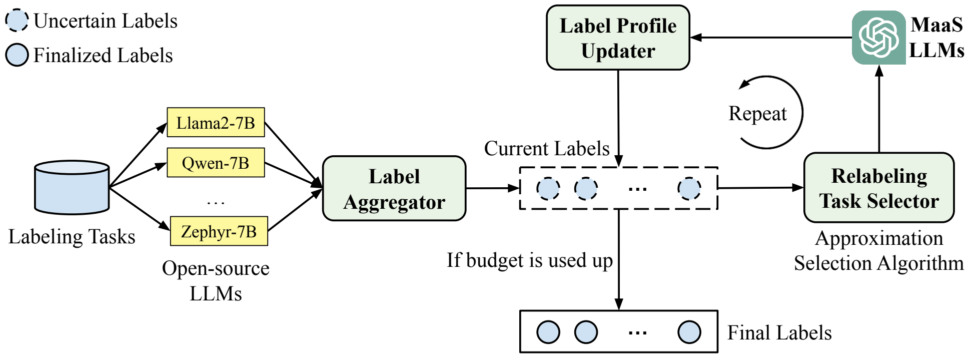


Figure 1: LMCrowd framework utilizes Free LLMs to initialize labeling for unmarked data and employs classical crowdsourcing aggregation algorithms to aggregate initial label distribution on the left. On the right, a selection-collection-update cycle is implemented, utilizing MaaS LLMs to inspect tasks based on our proposed algorithm, provide feedback, and update labels until the budget constrains are met.

a stochastic variable on $\{true, false\}^{T'}$. The expected quality of the data can only be calculated before crowdsourcing the result set from the LLMs.

**Definition 4 (expected quality)** *Given a data set $T$ and a query set $T'$, the anticipated data quality after the MaaS LLM relabels $T'$ is*

$$\mathbb{E}Q(T|T') = \sum_{A(T') \in \mathcal{A}(T')} P(A(T'))Q(T|A(T')) \quad (6)$$

where the $Q(T|A(T'))$ is the conditional data quality following receipt of a specific result set $A(T')$. We will introduce the computation of $Q(T|A(T'))$ in the next section.

We exploit the utility of free LLMs as crowd annotators to independently label the dataset. The label aggregator then merges these labels using a certain aggregation strategy. Despite the limitations of free LLMs' reasoning power, the aggregated result still contains significant noise. Hence, after aggregation, we utilize MaaS LLMs to verify selected labeled data, boosting data quality further.

### 3.2 Relabeling by MaaS LLMs

The use of MaaS LLMs entails a recurrent sequence of relabeling task selection, answer collection and label distribution update, which we shall henceforth refer to as a *round* (Figure 1). Because the relabeling task selection requires to calculate the expected data quality enhancement, we introduce the label distribution update first.

#### 3.2.1 Label Distribution Update

The relabeling queries and crowdsourced answers are still inherently uncertain, and merging the labels from MaaS LLMs with the profiles can be interpreted as as conditioning the posterior profile probability based on the relabeling results. In this scenario, Bayesian theorem can be effectively applied. It is worth noting that even when $\mathbf{r}$ and $A(T')$ agree on all the labeling queries and $T' = T$, i.e., the answers collected from the crowd are identical to the profile $\mathbf{r}$, $P(\mathbf{r})$ and $P(A(T'))$ may still differ. Suppose that the relabeling results for a query set $T'$ have already collected from MaaS LLMs, denoted as $A(T')$, they are then used to update the label distribution, namely the probability of each profile $\mathbf{r} \in \mathcal{R}$ from $P(\mathbf{r})$ to $P(\mathbf{r}|A(T'))$,

$$P(\mathbf{r}|A(T')) = \frac{P(\mathbf{r}) \cdot P(A(T')|\mathbf{r})}{P(A(T'))} \quad (7)$$

where the probability $P(A(T')|\mathbf{r})$

$$P(A(T')|\mathbf{r}) = \prod_{t \in cons(\mathbf{r}, A(T'))} Pr_{m_t} \cdot \prod_{t \in incons(\mathbf{r}, A(T'))} (1 - Pr_{m_t}) \quad (8)$$

The data will update based on the relabeling results from LLMs.

**Lemma 1 (label distribution update)** *For a relabeling task set $T'$, the collected result set from MaaS LLM $m$ is $A(T')$, the accuracy rate of $m$ is $Pr_m$, the label profile distribution is update by Equation 9.*

It is worth noting that LMCrowd framework supports using multiple MaaS LLMs, and the selected relabeling tasks can be distributed to different MaaS instances. If only one MaaS LLM $m$ is used for relabeling, the accuracy rate $Pr_{m_t} = Pr_m$ for each task $t \in T$.

#### 3.2.2 Optimal Task Selection for MaaS

The objective of the relabeling task selection is as follows.

**Definition 5 (optimal relabeling task selection)** *Given a task set $T$, along with possible profiles $\mathcal{R}$ that possess a joint distribution of probability and a LLM with diverse private accuracy $Pr_m$, our*

$$P(\mathbf{r}|A(T')) = \frac{P(\mathbf{r}) \cdot P(A(T')|o)}{P(A(T'))} = \frac{P(\mathbf{r}) \cdot \prod_{t \in cons(\mathbf{r}, A(T'))} Pr_{m_t} \cdot \prod_{t \in incons(\mathbf{r}, A(T'))} (1 - Pr_{m_t})}{\sum_{\mathbf{r}' \in \mathcal{R}} \Big( P(\mathbf{r}') \cdot \prod_{t \in cons(\mathbf{r}', A(T'))} Pr_{m_t} \cdot \prod_{t \in incons(\mathbf{r}', A(T'))} (1 - Pr_{m_t}) \Big)} \tag{9}$$

*objective is to optimize the expected data quality $\mathbb{E}Q(T|T')$ by choosing a size-k query set $(T')^*$ to pose to the LLMs,*

$$(T')^* = \underset{T' \subseteq T, |T'|=k}{\arg\max} \; \mathbb{E}Q(T|T') \tag{10}$$

Before discussing the task selection strategies, we would like to define the quality improvement after we get answers from LLMs. We denote the expected quality improvement as $\Delta\mathbb{E}Q(T|T') = \mathbb{E}Q(T|A(T')) - Q(T)$. Now, we formally present the core theoretical outcome of this study.

**Theorem 1** *The optimal selection of the relabeling tasks for MaaS LLM is to select the task set that maximizes the result set space entropy value,*

$$(T')^* = \underset{T' \subseteq T, |T'|=k}{\arg\max} \; \Delta\mathbb{E}Q(T|T') = \underset{T' \subseteq T, |T'|=k}{\arg\max} \; H(\mathcal{A}(T')) \tag{11}$$

Notice that $(T')^*$ is not necessarily the set of top-$k$ queries with the highest entropy values, namely, the following query set:

$$\underset{T' \subseteq T, |T'|=k}{\arg\max} \; H(T') = -\sum_{t \in T'} P(t) \log P(t)$$

is not necessarily equivalent with $(T')^*$. The optimization problem outlined above, when solved using Algorithm 1, also referred to as **Exact**, is NP-hard.

**Theorem 2** *Selecting the optimal query set of size $k$ from a data set of size $n$ is NP-hard.*

The proof is available in the Appendix A.1.

### 3.2.3 Approximation

Owing to the NP-hard complexity of the selection, the optimal solution cannot be obtained within polynomial time unless *NP=P*. Nevertheless, it has been demonstrated that conditional entropy is a submodular function (Krause and Guestrin, 2005). The challenge of selecting a k-element subset maximizing a monotone submodular function can be estimated with a performance bound of $(1 - 1/e)$ through the iterative chosen of the variable with the highest uncertainty relative to the already selected elements (Nemhauser et al., 1978).

**Theorem 3** *By posing uncertain labeling queries, the quality of the data is expected to improve monotonically.*

The proof is available in the Appendix A.2. In general, we introduce an iterative approach for selecting the $k$ tasks, outlined in Algorithm 2, referred to as the approximation algorithm (Approx. for short), which produces a $(1 - \frac{1}{e})$-approximate solution. The local quality improvement of adding $t$ to the query set $T'$ is represented as

$$g^{T'}(t) = H(A(T' \cup \{t\})) - H(A(T')) \tag{12}$$

Hence, the selection of the relabeling task set can be accomplished by incrementally select the local optimal relabel tasks,

$$t^* = \underset{t \in T \setminus T'}{\arg\max} g^{T'}(t) \tag{13}$$

until $|T'| = k$. In general, we propose three different task-selection algorithms in Task Selector: **Exact**, **Approx.** and **Random**. More details about the algorithm are available in the Appendix A.3.

## 4 Experimental Setup

### 4.1 Language Models

We select basic, general-purpose LLMs instead of those fine-tuned for specific tasks, and choose better open-source LLMs which are easy to deploy to ensure the accuracy of the initial markup model. Each model's accuracy can be measured using a test subset with ground truths. We use four open source universal LLMs: **LlaMa2-7B**, **FLAN-T5-base**, **Zephyr-7B**, and **Qwen-7B** from Hugging Face[1] and the best paid MaaS LLM at present **ChatGPT** (GPT-4 API) from OpenAI[2]. For each LLM, we used a 4-shot prompt to leverage the in-context learning capabilities of the model.

### 4.2 Label Agrregator

Our LMCrowd method is designed to be versatile enough to incorporate any "machine-only" fusion model that produces probabilistic results. To demonstrate its flexibility, we test it with eight different label aggregation methods: **BWA** (Li et al., 2019a), **CRH** (Li et al., 2014), DS (Dawid and Skene, 1979), **EBCC** (Li et al., 2019b), **GLAD** (Whitehill et al., 2009), **BCC** (Kim and Ghahramani, 2012), **MV** (Kittur et al., 2008), and **ZC**

[1] https://huggingface.co/

[2] https://platform.openai.com/

(Demartini et al., 2012). Further details about these baselines can be found in the Appendix A.5. In LMCrowd, the results collected from the free LLMs are merged with **MV** (Kittur et al., 2008) aggregation algorithm. Similar to the previous experiments, we employ the approximate selection algorithm and maintain a relabeled task set size of 2.

### 4.3 Baselines

We compare **LMCrowd** with two other data labeling strategies: manual labeling only (**Human**), GPT-4 labeling only (**GPT-4**). For manual annotation, in order to simulate a realistic scenario, we follow the charging standard of Google Cloud[3] and hire 3 university students and 6 graduate students to annotate the dataset.

### 4.4 Dataset

We perform a series of experiments on three datasets to assess the performance of our LMCrowd method. RTE (Wang et al., 2018) is a binary dataset of text implication tasks: entailment or not entailment. BoolQ (Clark et al., 2019) is a dataset for reading comprehension consisting of questions that require a yes or no answer. We sample up $1.5K$ cases from BoolQ for labeling. In COPA (Gordon et al., 2011), each question consists of one premise and two alternatives, where the task is to choose the alternative that is more causally related to the premise. Table 2 in Appendix A.4 shows the performance of different LLMs on these three datasets.

### 4.5 Tariff Analysis

We conducted a comparison of the cost breakdown between using MaaS GPT-4 and crowdsourced labels. The comparison details are presented in Table 1. Since we selected LLMs with the desired accuracy, we prioritized ease of deployment, so we disregarded the overhead of LLM selection. Additionally, to facilitate the comparison, we ignored the selection cost of GPT-4 prompts and the selection cost of crowdsourcing workers, and only considered the markup cost on the API or crowdsourcing platform. In the all subsequent experiment, we followed the same procedure. Tokens are the average number of tokens per piece of data in the data set. For manual labeling, each BoolQ question costs an average of \$0.303, each RTE question costs an average of \$0.111 and each COPA question costs an average of \$0.110. For GPT-4 using 4-shot prompt, the cost is $tokens \times 6 \times 10^{-5} \times 5$, where $6 \times 10^{-5}$ is the cost GPT-4 charged per token and 5 is the 4-shot.

Table 1: Tariff($) comparison of per GPT-4 and Crowdworker labeling

| Dataset | Tokens | Human($) | GPT-4 ($) |
|---|---|---|---|
| BoolQ | 117.4 | 0.303 | $1.767e-2$ |
| RTE | 43.0 | 0.111 | $0.651e-2$ |
| COPA | 41.6 | 0.110 | $0.63e-2$ |

## 5 Experimental Results

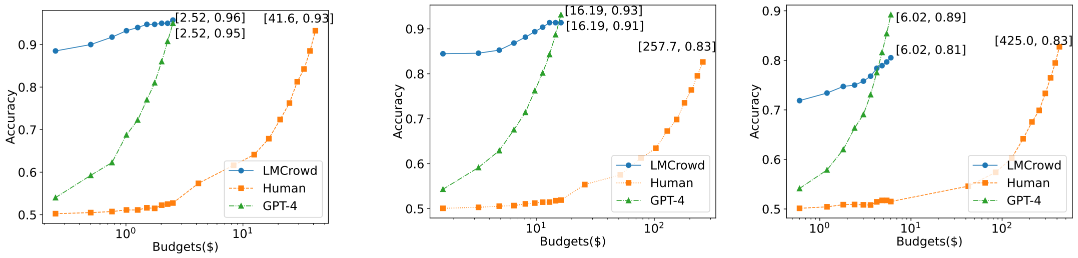
### 5.1 Primary Performance Analysis

Figure 2 presents an analysis of accuracy across various labeling strategies as budget increases. Our experimental results represent the maximum accuracy achieved using a combination of different labeling aggregator, task selector, and size of the relabeled task set. Specifically, for our main experiment, we employ the Approx. algorithm and adopt the MV (Kittur et al., 2008) aggregation strategy, while setting the size of the relabeled task set to 2. From the figure, it is evident that LMCrowd labeling outperforms single-source labeling with GPT-4 under a limited budget, and achieves significantly greater accuracy than human labeling. Of note, when aggregating the labeling results of four open-source models, LMCrowd yields an impressive peak accuracy of 87% on the COPA dataset. Additionally, on the remaining two datasets, LMCrowd attains accuracies exceeding 70%. As the budget allows for full coverage of GPT-4 costs, LMCrowd approaches and even exceeds GPT-4's fully-labeled accuracy, reaching 0.96 versus 0.95 on COPA in Figure 2(a). Importantly, LMCrowd's use of large models for labeling incurs significantly less expense than manual labeling while maintaining superior accuracy, resulting in a notable reduction in costs. Overall, LMCrowd delivers a substantial savings of up to 80% relative to fully GPT-4 labeling, and an exponential cost-savings compared to Human labeling.

### 5.2 Hyperparameter Study

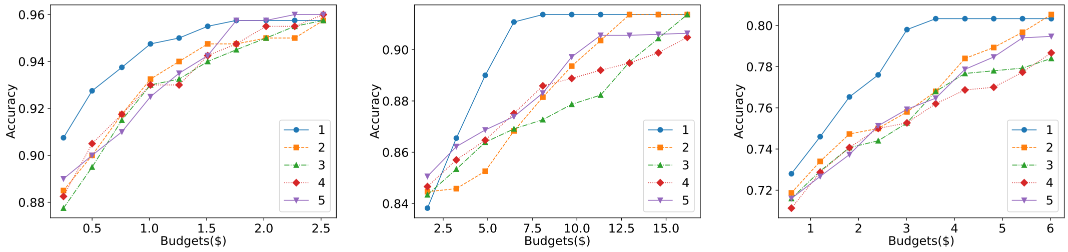
In accordance with our theoretical framework, it is imperative to carefully select the number of tasks, denoted as $k$, for LLMs to check. The selection of $k$ requires thoughtful consideration, as highlighted in Figure 3. Larger values of $k$ result in more checking tasks in each iteration, resulting in reduced interaction costs with LLMs under the same budget. As the budget increases, the rate of increase in accuracy decreases. When $k$ is exceptionally small,

[3] https://cloud.google.com/ai-platform/data-labeling/pricing?hl=en



(a) COPA (b) RTE (c) BoolQ

Figure 2: Accuracy Comparison with Baselines



(a) COPA (b) RTE (c) BoolQ

Figure 3: Hyperparameter study on different relabeling task set sizes

the improvement brought by each checking task in each iteration becomes imperceptible. Once the budget allows for full coverage of GPT-4 cost, the difference in accuracy between different $k$ values is at most $2\%$.

### 5.3 Ablation Study

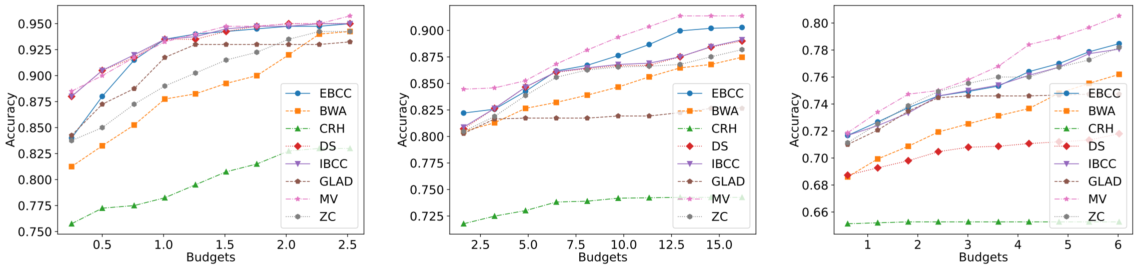
**Different Aggregators** As shown in Figure 2, we utilize the Approx. selection algorithm and maintain a task set size of 2, similar to previous experiments. Figure 4 shows that the MV aggregation method consistently outperforms the other seven alternatives across all three datasets, with a stable upward trend until reaching a plateau at full GPT-4 budget allocation. As the budget increases, the performance gap narrows, indicating overall improvement. However, the CRH method underperforms in our framework.

**Different Selection Methods** Besides, we demonstrate a comparison of 3 task-selection algorithms in Relabeling Task Selector. To evaluate the performance of these algorithms, we present the results in Figure 5 which illustrate the changes in accuracy for different values of $k$ (2, 3, and 4) in COPA dataset. Since according to Section 3.2.3, when k is 1, Approx. and Exact choose the same task. Our results show that when $k$ is 2, 3, or 4, the Exact and Approx. lines in the graph surpass Random, with Approx. approaching Exact, especially when $k$ is 4. Therefore, when larger values of $k$ make brute-force selection of tasks an NP-hard problem, we can opt for the Approx. method to approximately optimize the solution. Overall, our research provides meaningful perspectives into the performance of different task-selection algorithms and their suitability for different values of $k$.

## 6 Related Work

LLMs have revolutionized NLP and exhibit remarkable emergent capabilities without explicit programming (Wei et al., 2022). Furthermore, LLMs are capable of learning multiple NLP tasks simultaneously, achieving excellent performance on various tasks (Radford et al., 2019b). Given their impressive abilities, LLMs can be leveraged as effective crowdworkers for data labeling tasks, improving labeling quality within a limited budget through our proposed methodology.

Crowdsourcing can enhance data labeling quality through two primary methods. The first approach involves assigning tasks to multiple workers and integrating and deducing the correct result, but it is challenging due to the removal of false or irrelevant information from numerous responses. Label reasoning and incentive mechanisms (MacCartney et al., 2008; Gururangan et al., 2018) have been studied to address this issue. Result aggregation algorithms such as majority voting (Nitzan and Paroush, 1982), weighted voting (Tao et al., 2019), statistical knowledge like Naive Bayes (Zhang et al., 2023), and result collaboration via secondary crowdsourcing are typically used. The

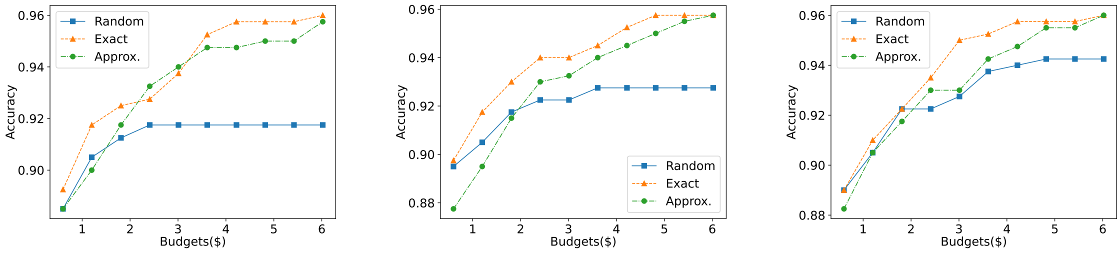


(a) COPA

(b) RTE

(c) BoolQ

Figure 4: Ablation Study on different label aggregation methods



(a) $k = 2$

(b) $k = 3$

(c) $k = 4$

Figure 5: Ablation Study on different relabeling task selectors in COPA

second approach involves cleaning the data with the highest noise rate based on the data noise rate to improve overall dataset quality (Ambati, 2012). The key challenge is to quickly access the most uncertain instance labels and verify them with experts. LMCrowd, our proposed approach, integrates both methods by assigning tasks to different LLMs as regular workers and combining and deducing their results, with LLMs used as experts for verification.

Selection strategies, such as active learning, help minimize the labeled data needed for machine learning tasks. For example, in (Settles and Craven, 2008), the Query by Committee (QBC) method identifies the most informative points, while a similarity-based method finds the most representative ones. Additionally, (Fang et al., 2014) introduces an active learning approach in crowdsourcing that selects the most informative examples and queries their labels from experts. We aggregate responses from multiple LLMs using label aggregation algorithms, calculate entropy from the initial labeled dataset's predicted probability distribution, and select the sample with the highest entropy for labeling, thereby reducing costs. There are studies focused on accurately labeling datasets while reducing costs. With the growing adoption of machine learning as a service (MLaaS) APIs, efficiently utilizing these services is crucial. FrugalMCT (Chen et al., 2022) optimally selects combinations of APIs based on data and budget constraints to achieve accuracy and cost trade-offs. However, the service industry is transitioning to Model-as-a-Service (MaaS). Our goal is to achieve accuracy and cost trade-offs by leveraging open-source LLMs while minimizing the use of paid APIs. In (Wang et al., 2021), GPT-3 serves as a data annotator, with logits from the API used as confidence scores, selecting the worst for manual labeling. Labeling costs with paid or partially automated models can be extremely high for supervised training datasets. We propose LMCrowd, which uses open-source models to initially label all data and selects items with the greatest impact for further verification by MaaS LLMs. LMCrowd can continuously label data faster than human clerks and at a much lower cost than using only MaaS LLMs, while maintaining accuracy.

## 7 Conclusion

We present a novel framework for efficient data labeling with LLMs under limited resources. We address the challenges of noisy labels and high costs associated with the task. By combining multiple free LLMs and MaaS LLMs, LMCrowds achieves improved labeling accuracy while minimizing expenses. Experimental evaluations across real-world datasets demonstrate the superiority of our approach over strong baselines. Our research pioneers the optimization of LLMs' usage with resource limitations, providing valuable insights into improving the quality and cost-effectiveness of data labeling processes.

## Limitations

We present a novel approach to hierarchical crowdsourcing and efficient data annotation leveraging a powerful large language model (LLM) called LMCrowd. This methodology ensures precise annotation within resource constraints. By comparing direct manual tagging with direct GPT-4 tagging, we have demonstrated that LMCrowd consistently outperforms baseline methods when budget considerations are prioritized. However, our research is limited to three foundational NLP datasets—RTE, BoolQ, and COPA, and a restricted range of generic LLMs. We acknowledge the need to expand our investigation to include a wider range of datasets and LLMs across diverse domains. Future endeavors can focus on evaluating the adaptability of our approach in various fields and exploring alternative large-scale LLMs architectures. Such endeavors promise to enhance the scope and effectiveness of automated data labeling methodologies.

## References

Vamshi Ambati. 2012. *Active learning and crowdsourcing for machine translation in low resource scenarios*. Ph.D. thesis, Carnegie Mellon University, USA. AAI3528171.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. 2020. Dataset search: a survey. *The VLDB Journal*, 29(1):251–272.

Lingjiao Chen, Matei Zaharia, and James Zou. 2022. Efficient online ml api selection for multi-label classification tasks. In *International conference on machine learning*, pages 3716–3746. PMLR.

Reynold Cheng, Jinchuan Chen, and Xike Xie. 2008. Cleaning uncertain data with quality guarantees. *Proceedings of the VLDB Endowment*, 1(1):722–735.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer errorrates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28.

Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, pages 469–478.

Meng Fang, Jie Yin, and Dacheng Tao. 2014. Active learning for crowdsourcing using knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Andrew Gordon, Cosmin Bejan, and Kenji Sagae. 2011. Commonsense causal reasoning using millions of personal stories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 1180–1185.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *Artificial Intelligence and Statistics*, pages 619–627. PMLR.

Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 453–456. ACM.

Saketh Kotamraju and Eduardo Blanco. 2021. Written justifications are key to aggregate crowdsourced forecasts. In *2021 Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021*, pages 4206–4216. Association for Computational Linguistics (ACL).

Andreas Krause and Carlos Guestrin. 2005. A note on the budgeted maximization of submodular functions. *SCS Technical Report Collection*.

Ziliang Lai, Chenxia Han, Chris Liu, Pengfei Zhang, Eric Lo, and Ben Kao. 2021. Top-k deep video analytics: A probabilistic approach. In *Proceedings of the 2021 International Conference on Management of Data*, pages 1037–1050.

Guoliang Li, Jiannan Wang, Yudian Zheng, and Michael J Franklin. 2016. Crowdsourced data management: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 28(9):2296–2319.

Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. 2014. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1187–1198. ACM.

Yuan Li, Benjamin IP Rubinstein, and Trevor Cohn. 2019a. Truth inference at scale: A bayesian model for adjudicating highly redundant crowd annotations. In *The World Wide Web Conference*, pages 1028–1038.

Yuan Li, Benjamin Rubinstein, and Trevor Cohn. 2019b. Exploiting worker correlation for label aggregation in crowdsourcing. In *International conference on machine learning*, pages 3886–3895. PMLR.

Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 802–811. ACL.

Winter Mason and Duncan J Watts. 2009. Financial incentives and the" performance of crowds". In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 77–85.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294.

Shmuel Nitzan and Jacob Paroush. 1982. Optimal decision rules in uncertain dichotomous choice situations. *International Economic Review*, pages 289–297.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019a. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019b. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, pages 1070–1079.

Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622.

Dapeng Tao, Jun Cheng, Zhengtao Yu, Kun Yue, and Lizhen Wang. 2019. Domain-weighted majority voting for crowdsourcing. *IEEE Transactions on Neural Networks and Learning Systems*, 30(1):163–174.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? gpt-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in neural information processing systems*, 22.

Chen Jason Zhang, Lei Chen, Mengchen Zhang, and Yongxin Tong. 2018. Reducing uncertainty of schema matching via crowdsourcing with accuracy rates. *IEEE Transactions on Knowledge and Data Engineering*, 32(1):135–151.

Chen Jason Zhang, Haodi Zhang, Weiteng Xie, Nan Liu, Kaishun Wu, and Lei Chen. 2023. Where to: Crowd-aided path selection by selective bayesian network. *IEEE Transaction of Knowledge and Data Engineering*, 35(1):1072–1087.

# A Appendix

## A.1 Proof of Theorem 2

**Proof A.1** *Selecting the optimal query set to enhance the desired query quality constitutes a pivotal optimization challenge. One approach involves exhaustively identifying all potential task sets, subsequently computing the spatial entropy of each resulting set, and ultimately selecting the task set with the highest entropy. Evidently, the time complexity of this method is exponential due to the necessity of considering all feasible subsets, making the problem NP-hard. To formally demonstrate this, the problem can be reduced to a recognized NP-hard problem, such as the set coverage problem. This problem involves identifying a subset of S from a set U of M elements and a set S of N subsets. The goal is to include all elements of U with the smallest possible size.To reduce this problem to the set covering problem, each data point in the dataset is treated as an element in set U. All data points in the dataset are then considered as a large set S, where each subset represents a potential subset. By calculating the entropy of the result set space for each subset, where each data point in the subset is treated as an element in set U, the entropy of the result set space for the subset represents its size. The ultimate goal is to identify a subset that maximizes the entropy of the result set space, which is equivalent to finding a subset in the set coverage problem with minimal size. Given that the set covering problem is NP-hard, determining the optimal set of queries as stated in Theorem 2 is also NP-hard. This inferential approach fosters a deeper comprehension to address the optimization problem and provides guidance for subsequent algorithmic design.*

## A.2 Proof of Theorem 3

**Proof A.2** *Given a task set* $\mathcal{T}$ *and a query* $t \notin \mathcal{T}$ *such that* $0 < P(t|\mathcal{T}) < 1$ *the quality gain* $g^{\mathcal{T}'}(t) = H(\mathcal{AS}^{\mathcal{T}'}) - H(\mathcal{AS}^{\mathcal{T}}) = H(A_{m_t}^{\mathcal{T}'}|\mathcal{T})$*, where* $\mathcal{T}' = \mathcal{T} \cup t$*. As each individual task is approached and addressed independently, we have* $H(A_{m_t}^{\mathcal{T}'}|\mathcal{T}) = H(A_{m_t}^{\mathcal{T}'})$*. Since* $f \notin \mathcal{T}$ *and the query* $t$ *is an uncertain task, i.e.* $0 < P(t|\mathcal{T}) < 1$*, we have,*

$$\begin{aligned} 0 <& P(A_{m_t}^{\mathcal{T}'} = True) \\ &= Pr_{m_t} \cdot P(t|\mathcal{T}) + (1 - Pr_{m_t}) \cdot P(\neg t_j|\mathcal{T}) < 1. \end{aligned} \tag{14}$$

*It is noteworthy that* $H(A_{m_t}^{\mathcal{T}'})$ *attains a positive value. Consequently, posing a query* $t$ *will lead to an enhanced quality of the data.*

### A.3 Selection Methods

We propose three distinct task-selection algorithms for the relabeling task selector utilized in LMCrowd.

**Algorithm 1** Exact Relabeling-Task Selection

```
Input: Task set $\mathcal{T}$, $k$ and budget $B$
Output: Task set $\mathcal{T}'$
Initialize $\mathcal{T}' \leftarrow \emptyset$
Initialize $max \leftarrow 0$
for each $k$ elements subset $\mathcal{T}^*$ of $\mathcal{T}$ do
    Set $g^{\mathcal{T}^*}(t) \leftarrow H(A(\mathcal{T}^*)) - H(A(\mathcal{T}))$
    if $g^{\mathcal{T}^*}(t) > max$ then
        Set $max \leftarrow g^{\mathcal{T}^*}(t)$
        Set $\mathcal{T}' \leftarrow \mathcal{T}^*$
    end if
end for
return $\mathcal{T}'$
```

**Algorithm 2** Approx. Relabeling-Task Selection

```
Input: Task set $\mathcal{T}$, $k$ and budget $B$
Output: Task set $\mathcal{T}'$
Initialize $\mathcal{T}' \leftarrow \emptyset$
repeat
    select $t = \arg\max_{t \in \mathcal{T}} H(A(T' \cup \{t\})) - H(A(T'))$
    if $H(A(T' \cup \{t\})) - H(A(T')) \leq 0$ then
        break
    else
        Set $\mathcal{T}' \leftarrow \mathcal{T}' \cup \{t\}$
        Set $\mathcal{T} \leftarrow \mathcal{T} \setminus \{t\}$
    end if
until $|\mathcal{T}'| = \min(k, B)$
return $\mathcal{T}'$
```

**Exact** For Algorithm 1, the selection of the optimal query set containing the top-k queries with the highest entropy values involves a comprehensive examination of all possible subsets to determine it.

**Approx.** To tackle the inherent NP-hardness of Exact, we employ the algorithm as an approximate solution. As outlined in Algorithm 2, our approach involves selecting the task that can yield the maximum increase in entropy value at each step, and adding it to the subset $\mathcal{T}'$ until either $\mathcal{T}'$ contains $k$ elements or the budget has been fully utilized.

**Random** In each new task of the aggregate, randomly select k queries to check.

### A.4 Details of the LLMs Used in Our Method

- **LlaMa2-7B**[4]: LlaMa2-7B, released by Meta, is a language model with 7 million parameters. Trained on extensive text datasets, LlaMa2-7B possesses robust natural language processing capabilities. It is versatile and can be utilized across various tasks, such as question answering, text summarization, and text generation.

- **FLAN-T5-base**[5]: This language model, developed by Google, operates on the T5 architecture and comes pre-trained. Trained on a substantial corpus of text data, it possesses versatile capabilities suitable for numerous NLP tasks. These tasks encompass, but are not confined to, question answering, text summarization, and text generation.

- **Zephyr-7B**[6]: This formidable language model boasts 700 million parameters and comprises a complex network of deep neural networks. Trained extensively on vast datasets, it exhibits proficiency in comprehending, generating, and reasoning with natural language. Its versatility makes it apt for tackling a myriad of text processing tasks.

- **Qwen-7B**[7]: This language model, equipped with up to 700 million parameters, is designed to deliver swift and effective natural language processing solutions. It undergoes pre-training on extensive corpora and further refinement through fine-tuning across various language understanding tasks. With its broad spectrum of capabilities, including text generation, language comprehension, and sentiment analysis, it holds significant potential for diverse applications.

Table 2: LLM Performance (Accurary)

| LLMs | BoolQ ($) | COPA($) | RTE ($) |
|---|---|---|---|
| Zephyr-7B | 0.7993 | 0.7775 | 0.7920 |
| Qwen-7B | 0.8120 | 0.7775 | 0.7277 |
| LlaMa2-7B | 0.8193 | 0.6725 | 0.7144 |
| FLAN-T5-base | 0.7280 | 0.7800 | 0.8229 |
| GPT-3.5 | 0.787 | 0.8458 | 0.8462 |
| GPT-4 | 0.892 | 0.9525 | 0.9317 |

[4] https://huggingface.co/meta-llama/Llama-2-7b

[5] https://huggingface.co/google/flan-t5-base

[6] https://huggingface.co/HuggingFaceH4/zephyr-7b-beta

[7] https://huggingface.co/spaces/Qwen/Qwen-7B-Chat-Demo

### A.5 Details of the Aggregation Methods

Our LMCrowd method is engineered with flexibility in mind, capable of seamlessly integrating any "machine-only" fusion model generating probabilistic outcomes. To showcase its adaptability, we subject it to scrutiny with eight distinct label aggregation methods.

- **EBCC** (Expectation-Based Crowd Counting) (Li et al., 2019b): EBCC addresses the counting problem by considering the relationships among participants and using the expectation maximization algorithm to derive results. While effective, it requires prior knowledge and can be computationally intensive due to the iterative nature of expectation maximization.

- **BCC** (Bayesian Crowd Counting) (Kim and Ghahramani, 2012): BCC also targets counting problems and employs Bayesian inference to estimate worker abilities and answer distributions, using these estimates to deduce counting results. The computational cost can vary, depending on the complexity of the Bayesian model used.

- **BWA** (Bayesian Worker Aggregation) (Li et al., 2019a): BWA utilizes Bayesian inference to estimate participant abilities and answer distributions, which are then used for answer aggregation. Its computational cost is moderate, as it relies on inference techniques that can scale with the number of participants.

- **CRH** (Crowd-based Relational Hierarchical mode) (Li et al., 2014): CRH focuses on relational data and uses participant relationships to deduce answers through a hierarchical model. While it can provide nuanced results, the hierarchical structure may lead to higher computational costs compared to simpler models.

- **GLAD** (Generative model for Labeling Aggregation with Diagnostics) (Whitehill et al., 2009): GLAD is based on participant ability and answer accuracy, employing a generative model to estimate both. This method can be computationally demanding due to the need for complex estimations.

- **ZC** (Demartini et al., 2012): This method utilizes deep learning techniques to integrate answer aggregation and participant ability prediction to form an end-to-end system. ZenCrowd employs a neural network-based model that automatically learns the abilities of the participants and the distribution of the answers, and aggregates the answers. While powerful, the use of neural networks can result in significant computational costs, especially during training.

- **DS** (Dawid-Skene) (Dawid and Skene, 1979): DS derives answers by considering individual abilities and biases. It first predicts these parameters and then computes a weighted average of answers. The computational cost is generally lower than that of GLAD, but it can still be significant due to the initial prediction phase.

- **MV** (Majority Voting) (Kittur et al., 2008): MV operates on the principle of majority voting, where all participant answers are counted, and the most frequent answer is taken as the final result. This method is computationally inexpensive, making it a popular choice for many applications.