On the impact of predicate complexity in crowdsourced classification tasks

Jorge Ramirez†, Marcos Baez○, Fabio Casati‡, Luca Cernuzzi○, Boualem Benatallah△, Ekaterina A. Taran‡, and Veronika A. Malanina‡

†University of Trento, Italy. ○Université Claude Bernard Lyon 1, France. △ServiceNow, USA.
‡Catholic University Nuestra Señora de la Asunción, Paraguay. ‡University of New South Wales, Australia. ‡Tomsk Polytechnic University, Russia.

ABSTRACT
This paper explores and offers guidance on a specific and relevant problem in task design for crowdsourcing: how to formulate a complex question used to classify a set of items. In micro-task markets, classification is still among the most popular tasks. We situate our work in the context of information retrieval and multi-predicate classification, i.e., classifying a set of items based on a set of conditions. Our experiments cover a wide range of tasks and domains, and also consider crowd workers alone and in tandem with machine learning classifiers. We provide empirical evidence into how the resulting classification performance is affected by different predicate formulation strategies, emphasizing the importance of predicate formulation as a task design dimension in crowdsourcing.

KEYWORDS
crowdsourcing, task design, predicate complexity

1 BACKGROUND & MOTIVATION
Micro-task crowdsourcing today is still an art. Indeed, it is not surprising that companies charge hefty consulting fees to help businesses set up and run crowdsourcing tasks. Successful projects involve designing and harmonizing several aspects, from designing the user experience to task design, training and test settings, and to seemingly easy problems such as how to ask questions and elicit truthful, accurate answers [12] — all while meeting budget constraints and treating your workforce fairly and with respect.

For example, longer instructions affect the task uptake by workers by three times, while showing concrete solution examples improves accuracy up to ten times [51], depending on the type of task. Mechanisms to combat task spammers are often essential, since without them a task can easily get half of the answer as invalid even on simple tasks, although it raises the contributors’ efforts (and cost) [26]. Budget is also a limiting factor, and reward strategies and optimization can also affect the results [2, 20, 29, 47]. The list is almost endless, so much that it is motivating crowdsourcing researchers to prepare design and reporting guidelines for crowd experiments [39].

This paper explores and provides guidance on a specific but important aspect of crowdsourcing task design: how to ask “complex” questions to the crowd to classify items. Classification in general is by far the most popular type of crowdsourcing tasks1. In this paper we study classification in the context of information retrieval and multi-predicate classification problems, that is, tasks where the crowd has to select items that meet a set of conditions. The “complexity” of the question comes therefore from the fact that it is composite, and we want our crowd worker to state if items satisfy our set of conditions (predicates). This is a very common task we do implicitly or explicitly countless of times in our daily life and that often appears in crowd tasks as well (from selecting hotels that have certain characteristics of interest [10] to screening papers for systematic literature reviews [47]). Indeed, any conjunctive query is an instance of such problem and abundant prior research on crowd query processing studied how to efficiently retrieve items from a potentially large set [5, 16, 34].

We tackle this problem because it is common enough to be of widespread interest and nuanced enough (as we show in this paper) to require a detailed investigation, and it can be framed so that it can result in reusable knowledge for task designers. In particular, we set to study the following research question:

How does the way we ask a composite question impacts the individual and aggregate performance of crowd workers?

We investigate the question both in the context of crowd-only classification and in hybrid classification, an increasingly common approach where humans and machines work together to solve a classification problem. We analyze the problem based on both characteristics of the question and of the task, such as task “length” (e.g., length of the document to read for text classification tasks), task domain, task difficulty, and class balance.

Surprisingly, the crowdsourcing literature somewhat overlooked predicate complexity in classification tasks. First, complex predicates may require longer task instructions, which is known to correlate positively with the perceived complexity (as seen by workers [53]), impact task intake (most workers tend to quit after inspecting the instructions [15]), and, therefore, the latency. Second, increased task complexity naturally demands more effort from workers, challenging accurate and fair compensation [50]. Last, task complexity plays an important role in the quality of the results obtained [14, 28].

The main contributions of this work are as follows. We introduce the problem of predicate formulation for crowd classification tasks as a relevant design dimension (to enhancing worker performance and enabling Human-AI collaboration). We study complexity in classification problems on a broad landscape of tasks considering

1A relatively recent worker survey on Appen, previously Figure Eight, shows that 45% of jobs are classification tasks [23]. Also, 60% of the built-in templates offered by Amazon Mechanical Turk constitute classification tasks, and 40% in Yandex Toloka.
categorization and classification, verification, content moderation, and sentiment analysis tasks (see [23] for a taxonomy of task types in crowdsourcing). Our experiments, therefore, cover multiple domains and leverages human and machine classifiers. We provide empirical evidence on the impact of predicate formulation on classification outcomes, suggesting performance gains when querying complex predicates as multiple simpler questions. We also provide insights into the expected performance of different formulation strategies under different i) problem settings such as predicate selectivity and class distribution, and ii) task design choices such as querying predicates on the same or separate tasks. The experiments also offer preliminary evidence on the potential of predicate formulation in the context of hybrid classification, suggesting performance gains even in its simplest collaborative approach, by assigning crowd and machine parts of a complex predicate they are more suited to classify. Last but not least, we contribute datasets derived from our experiments.

2 RELATED WORK
Task design in crowdsourcing
Task design is a multi-dimensional problem with a rich body of work in the crowdsourcing literature [6]. "Design" does not only mean the actual task interface, but also the mechanisms to deploy, coordinate, and assign tasks to workers, the tools to assure high-quality contributions, and budget management [12]. The lessons learned from this literature spawn on best practices for designing effective tasks (given the impact task design has on the resulting performance), and methods for performing crowdsourcing studies.

Crowdsourcing results are sensitive to subtle changes in task design. Poor instructions may lead workers to misinterpret the task and produce subpar responses [51]. The clarity of the task [24] and how it is framed (whether meaningfully or not) [1] may also swing workers’ performance. The prevalence of malicious workers in platforms asks for design decisions that account for this and guard quality (e.g., equip tasks with mechanisms to combat spammers [26]). Similarly, task design could aid worker performance, in the form of assistance to workers [38, 45], proper compensation for effort-intensive tasks [8], or by rigorous training protocols [7] and feedback loops [19]. Latency also matters and can be affected by ineffective instructions causing task abandonment [15] or generating mistrust in task requesters [27]. However, fair compensation can help to speed up task intake and how much workers contribute [8].

These lessons provided valuable insight into properly designing and running crowdsourcing studies. As design choices may swing the results obtained, it can also affect the validity of experimental outcomes [26]. Choices in task design can amplify biases inherent to crowdsourcing environments. Task clarity influences how workers pick tasks and, therefore, introduce selection effects [24]. The active pool of workers varies as hours go by [9], and with this, different decisions affecting when a crowdsourcing job runs could result in unanticipated performance differences and confounding factors [36]. The lack of built-in support from crowdsourcing platforms makes it difficult to run controlled experiments, making simple between-subjects design a challenging endeavor [26]. A common approach involves identifying workers via browser fingerprinting [22] and then using an external server to randomize participants to experimental conditions [38]. These challenges motivated the research community towards developing guidelines for designing and reporting crowdsourcing experiments [35, 39].

Multi-predicate classification
We study predicate formulation in the context of problems regarded as finite pool classification [33], where we have a finite set of items to classify according to a set of criteria (potentially) unique to the problem. Systematic literature reviews are one instance of this problem, and have been heavily studied in the crowdsourcing literature [11, 32, 46, 48]. Mortensen and colleagues [32] tested the feasibility of leveraging crowdsourcing, given the costs associated with producing SLRs [47]. They found that task design plays a major role in the quality of the results, as well as this can vary from predicate to predicate. Krivoshev et al. [11] proposed models and algorithms to crowdsource SLRs, offering quality and budget trade-offs to guide how to invest in the crowdsourcing tasks. Budget limits entire crowdsourced solutions, works have also focused on leveraging machine classifiers in tandem with crowd workers [29, 47]. For example, leveraging strategies such as classifying "easy" items first with ML and crowd for the rest [47] or modeling tasks and workers to determine promising predicates to filter out items.

Multi-predicate classification is also studied in the context of information retrieval. A common problem is to determine an optimal order of the predicates (to query the crowd for labels) to filter out tuples [5, 10, 42, 49]. Similarly, work in crowd-powered databases studied how to leverage crowdsourcing to extend the capabilities of database systems to answer complex multipart queries over flexible (or on-demand) schemas [16, 34].

Despite the vast body of work on task design and on information retrieval / multi-predicate classification, to the best of our knowledge, we are the first to study the impact on how the (complex) information retrieval question is formulated, a dimension that affects all of the prior art. Our experiments, over an ample range of tasks, emphasize the importance of the predicate formulation as a problem, and show its impact on classification outcomes.

3 PROBLEM AND APPROACH
We now define and scope the crowdsourced classification problem, and in Section 5, we introduce the crowd-machine variant.

The task we seek crowd help for is to identify all items in a set \( I \) that meet a complex predicate \( \mathcal{P} \), defined as the conjunction of predicates \( \{ p_1, p_2, \ldots, p_n \} \). For example, taking a common problem from the literature (screening scientific papers [11, 47]), \( \mathcal{P} \) could be a set of scientific articles returned by a keyword-based query on Scopus, and we may seek papers reporting experiments on older adults living in Africa (\( \mathcal{P} = p_1 \land p_2 \), where \( p_1 \): "Is the study population 65+ years?" and \( p_2 \): "Is the population living in Africa?".). To solve this problem, we have to our disposal a set of crowd workers \( W \), a budget \( B \), and a quality goal (or loss function) \( L \) to meet.

The predicate formulation problem seeks to determine how to ask the question in the context of multi-predicate classification. There are different ways to formulate a complex predicate, and, in this paper, we study specifically three ways: i) ask the complex question (e.g., "Is the study on 65+ years old adults living in Africa?"); ii) break the composite question into component predicates, but...
ask them as part of the same task, and iii) make each predicate a task of its own (which also means that a crowd worker only sees one predicate and assesses many items for it).

We approach this problem systematically, considering both characteristics of the question and tasks. To give breadth to our analysis, we explore a broad landscape of tasks (categorization and classification, verification, content moderation, and sentiment analysis tasks) representing different domains and task difficulty levels. We focus our experiments on document retrieval (text classification) and consider documents of different lengths, given the associated effort incurred on workers to (understand and) assess text, and the potential influence of the documents’ length on performance [14, 38]. Our focus on text stems from the fact that it is a recurrent use case in the literature [8, 11, 45], and annotating images are deemed simpler in comparison to annotating texts [28]. Finally, crowdsourcing tasks are prone to worker biases [21], which could be caused by frequently assigning items to the same class. Therefore, we consider different class distribution scenarios to study the predicate formulation problem in crowdsourcing contexts.

4 CROWDSOURCING EXPERIMENT

This experiment studies the impact of the task design alternatives on the performance of crowd workers. We focus on the simpler case where a complex predicate is composed of two simpler ones. We show the individual and collective performance gains related to predicate reformulation, and how the nature of the problem influences the resulting performance.

Datasets. We considered datasets with different characteristics in terms of domain, predicates, document length, and difficulty (classification accuracy), in line with prior art [29, 38]. The datasets come from systematic literature reviews (SLRs), customer feedback analysis, content moderation and crowd verification, and are representative of multi-predicate screening problems from the literature [29, 37, 52]. See the supplementary material\(^3\) for details on the predicate composition for each of the reference datasets.

Virtual reality exergames. This dataset was produced and annotated by the authors as part of their investigation into overlaps between SLRs. We identified a pool of 80K scientific articles from multiple SLRs that share some predicates. From this pool of papers, we built the Exergame-VR dataset that consists of 500 articles from 4 SLRs with high overlap in terms of predicates and papers within their scope. Additionally, we split the documents into two buckets based on their length: short (150 items with length \(\leq 230\) words) and long (350 items with length > 230 words).

Amazon product reviews. This dataset contains 100k reviews of products that are sold in Amazon [29]. It is labeled according to the following two predicates: 1. Book: “Is it a book review?” and 2. Negative: “Is it a negative review?” We randomly selected 236 reviews (118 short, and 118 long) to create the AMZ-reviews dataset.

Wikipedia detox. This dataset from Wulczyn et al. [52] contains 100k comments from “Talk pages” in Wikipedia, labeled by crowd workers on whether each of the comments contains a personal attack (or an attack of another kind). From this pool of 100k items, we built Content-Moderation, a dataset of 118 long documents (comments with > 230 words) labeled on two predicates.

Verifying crowd contributions. In [37], the authors contribute datasets where workers provided a binary label to a relevance question, and a highlighted excerpt to justify the labeling. We built the Verification dataset based on [37], selecting 118 long documents labeled according to predicates that determine whether the judgment and highlighted passage are correct. These tasks are relevant to iterative workflows, where workers act as reviewers [13].

Economic inequalities in older adults. This dataset is part of an ongoing SLR on assessing the inequalities in older adults. It contains 2619 papers. From this pool of documents, we selected and labeled 151 items to build Inequality-OA, a dataset of long abstracts.

Design. The task performed by workers in our experiment consisted of reading a piece of text and answering one or two binary questions of different complexity levels depending on the task design. Figure 1 shows an example of a task (inspired by prior art [11, 41]).

---

\(^3\)https://tinyurl.com/simpler-predicates-supp
cases\footnote{Representing the two predicates in each dataset as $p_1$ and $p_2$, the three exclusion cases are 1) $p_1 = 1$, $p_2 = 0$; 2) $p_1 = 0$, $p_2 = 1$; 3) $p_1 = 0$, $p_2 = 0$.}. As the name suggests, the 80-20 case represents a setting with roughly 20% of items included and 80% excluded. This skewed setting tends to be problematic in crowdsourcing since it may bias workers towards the most frequent answer\cite{21}. For this reason, and quality control purposes, the training and control items follows a 30 IN and 70 OUT distribution, making sure that each page of work shows items from both classes.

We consider four experimental conditions for our crowdsourcing experiments, each condition represents a variation of the task interface shown in Figure 1. The baseline condition we use as control, and it asks workers a complex predicate $\mathcal{P}$ (a question that integrates both predicates in a dataset, as indicated in 1). The $P_1$ - $P_2$ condition represents the task alternative that asks the constituents of $\mathcal{P}$ on the same task. The conditions $P_1$ and $P_2$ represent tasks that asks workers only one simpler predicate (predicates $p_1$ and $p_2$, respectively).

Initially, we considered both short and long documents. However, in a pilot study, we observed that the task alternatives did not improve over the baseline when considering short documents, suggesting that these may be more suitable for tasks where workers face longer documents\cite{38, 45}. Therefore, we consider only long documents in our study. We followed a between-subject design, assigning workers to one of the four experimental conditions. Workers judged a maximum of 18 documents that we divided into 6 pages of 3 items per page (6 x 3 design), and we only allowed workers that understand English. We required workers to perform a training task (one page of work) before advancing to the main task, a quality control mechanism typically done in crowdsourcing research\cite{31}. Workers that scored 100% advanced to the main task, where we included control items as an additional quality assurance mechanism. We required workers to maintain an accuracy level of 100% for the AMZ-reviews dataset and 76% for the rest of the datasets\footnote{Prior art\cite{29} shows that the baseline performance was quite high for AMZ-reviews; therefore, we defined the 100% quality threshold for this dataset.}. We paid workers between $0.09 and $0.21 per page of work (depending on the condition and dataset), aiming at an hourly rate of 7.5 USD. We collected contributions from workers on the Yandex Toloka platform\footnote{https://toloka.yandex.com/}, asking 3 votes per item in the datasets. We defined a timeframe from 14:00 to 21:00 GMT+1 for running the experiment, running each dataset separately with a time gap between these. We executed each of the experimental conditions in parallel and balanced the contributions from each geographical bucket (~33% per bucket within each condition).

We inspected the demographics of Toloka and noticed that roughly 90% of workers come from Russian-speaking countries, where Russia and Ukraine contribute the majority of the workers (~79% and ~10% respectively). Besides only allowing workers that understand English, we decided to create 3 geographic buckets: Russia, Ukraine, and the “Rest of the world”, balancing the contributions from these buckets in our experiments to avoid any bias due to demographics.

We used an external server to assign workers to experimental conditions in a round-robin fashion, blocking workers from jumping between conditions (to avoid learning effect). We added a custom JavaScript code to the task interface to call the external server and render the experimental condition accordingly\cite{40}.

4.1 Results

We collected a total of 8250 judgments from 1185 workers across the datasets we considered in this experiment. Here we describe our results to determine the impact of asking the complex predicate $\mathcal{P}$ vs. leveraging its simpler constituents on the classification performance of crowd workers.

Worker accuracy

We use the ground-truth labels available in the datasets to determine the classification accuracy of workers in each of the experimental conditions. The median accuracy of workers in the baseline was 0.89 for AMZ-reviews and 0.67 for the rest (Exergame-VR, Content-moderation, Verification, and Inequality-OA), it would seem that workers found it easier to judge product reviews than documents from the rest of the domains.

We test the significance of the difference in worker accuracy using the Kruskal-Wallis H test\footnote{In our pilot study, we noticed that the observations do not follow a normal distribution}. The test indicates that there is a significant difference between the experimental conditions in 4 out of 5 datasets ($p < 0.05$ for Exergame-VR, and $p < 0.01$ for the rest). The results are depicted in Figure 2. We analyze all possible pairwise comparisons using the Dunn’s Test of Multiple Comparisons\cite{4}, using Benjamini-Hochberg correction to reduce the probability of Type I error. It can be noted that either $P_1$ or $P_2$ has a significant improvement over the baseline (3 out of 5 datasets). And that the $P_1$-$P_2$ condition significantly outperforms the baseline in the Content-moderation and Inequality-OA datasets.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2.pdf}
\caption{Worker accuracy by experimental condition for the 60-40 case. The lines indicate significant differences, coding p-values as \*: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$, ****: $p \leq 0.0001$.}
\end{figure}

In the skewed class distribution scenario, the 80-20 case, the median worker accuracy in the baseline condition was 0.67 for both Exergame-VR and Inequality-OA (figure omitted due to space...
We analyze the results from a collective perspective and evaluate we introduce $P$ we are likely to see an increase in worker accuracy in at least one of the simpler predicate. Furthermore, by asking more granular and simpler predicates we obtain valuable detailed information about crowd and task characteristics. For example, according to our results, workers were better at evaluating if a review was about a book than whether it was a negative review (simple verification vs. sentiment analysis tasks), suggesting different difficulty levels. This detailed information could equip crowd-machine algorithms to make better decisions about what to crowdsource and what to automate.

**Classification performance**

We analyze the results from a collective perspective and evaluate the impact of predicate formulation in the resulting classification.

The overall classification, for a given $P = \{p_1, p_2\}$ in our datasets, is derived from the conjunction of the aggregated results from each of the simpler predicates (i.e., $p_1 \land p_2$ for each item $i \in I$). We use majority voting to aggregate the contributions from multiple workers and the $P_1$ score to assess the classification quality. The baseline condition already provides classification on $P$ since the simpler predicates are combined in a single question, and for the $P_1$-$P_2$ condition we simply take the conjunction of $\{p_1, p_2\}$. To make the results from $P_1$ and $P_2$ comparable to previous conditions, we introduce $P_1 \land P_2$, which also takes the conjunction of each simpler predicate. To compute the $P_1$ score for these conditions we use as ground-truth label the conjunction of the simpler predicates.

Table 1 summarizes the classification performance for the experimental conditions across our five datasets for both 60-40 and 80-20 cases. The baseline performance ranges between 0.6 (Exergame-VR) and 0.909 (AMZ-reviews).

We compared the performance of asking $P$ directly vs. asking the simpler predicates first and then combining the results ($P_1$-$P_2$ and $P_1 \land P_2$). It can be observed that the conditions $P_1$-$P_2$ and $P_1 \land P_2$ outperformed the baseline condition but not consistently across all datasets. $P_1$-$P_2$ outperformed the baseline in 3 out 5 datasets (Content-moderation, Verification, and Inequality-OA), with an increase in performance of up to 18%. $P_1 \land P_2$ improved over the baseline in 2 out 5 datasets, with an increase of up to 9%. In the 80-20 case, the $P_1$-$P_2$ condition showed superior performance when compared to the baseline, with an increase of up to 27% (while the $P_1$ & $P_2$ fell behind the baseline).

We also compared the simpler predicates against the complex one. We observed superior classification results when formulating a composite predicate as multiple (more straightforward) questions leveraged on the same or separate tasks, even when votes are aggregated with simple majority voting. Asking two simple questions on the same task (the $P_1$-$P_2$ condition) resulted in performance gains ranging from 6% to 48%. And the conditions $P_1$ and $P_2$ that asked a simple question surpassed the baseline performance in all datasets, with an increase in $P_1$ score ranging from 2% and up to 47% In the 80-20 case the $P_1$ scores in the baseline were 0.571 for Exergame-VR, and 0.476 for Inequality-OA. In both datasets, task formulating simple predicates outperformed complex ones (when delivered separately or together on the same task), with an increase in classification performance of up to 97%.

A closer look into the performance on the complex predicates (baseline condition) across the two class distribution scenarios showed that overly skewed datasets may hurt the classification performance of the crowd — $F_1$ decreased 4% for Exergame-VR, and 31% for Inequality-OA. While by leveraging simple predicates, the classification performance could remain roughly the same, except for the unusual case of $P_1$ for Inequality-OA, where the performance decreased 20%. We believe the selectivity of $P_1$ (see our supplementary material) played a role in this drop in performance since it is equal to 0.44 in the 60-40 version of Inequality-OA and 0.20 in the more skewed variant (also observed in the baseline).

In summary, there is evidence suggesting that complex multipart questions may benefit from disentanglement into simpler elements. As we observed, performance boosts can be obtained by formulating and presenting complex predicates as simple and more granular questions and combining back the results. However, there is no clear pattern for when each task design alternative (presenting simple predicates on the same- or separate tasks) will be the appropriate one to implement, an interesting direction for future work.

**Worker effort**

Although our main focus in this paper is quality, we complement our analysis by looking at the impact on worker effort. We consider *decision time* as a proxy to estimate the effort incurred on workers.

The median decision time in the baseline condition was 22.66s for Exergame-VR, 33.88s for AMZ-reviews, 30.77s for Content-moderation, 23.38s for Verification, and 25.59s for Inequality-OA. In the 80-20 datasets, the median decision time in the baseline condition was 33.62s for Exergame-VR, and 33.05s for Inequality-OA.

While formulating complex predicates as simpler multipart questions offer gains in quality, it results in slower task completion time. Workers in the $P_1$-$P_2$ condition spent significantly more time than the baseline in all datasets ($p < .01$), which intuitively makes sense since workers answered two questions rather than one (the decision time ranged between 36.77s and 53.56s). Likewise, the $P_1$ & $P_2$ condition was also significantly slower than the baseline ($p < .01$, with decision time between 36.96s and 45.08s)\(^8\). Also, there is no substantial evidence to suggest which task alternative ($P_1$-$P_2$ or $P_1 \land P_2$) is better in terms of effort. The conditions $P_1$-$P_2$ and $P_1$ & $P_2$ had comparable results in 3 out of 5 datasets ($p > .05$), and $P_1$ & $P_2$ outperformed on the rest ($p < .05$).

Looking closer into the performance, we noticed that simpler predicates (when viewed in isolation) could potentially be faster than asking a complex predicate, but not always. When comparing $P_1$ and $P_2$ to the baseline, we noticed two competing observations. One of the simpler predicates was significantly faster than the baseline in some cases (40% faster for AMZ-reviews, 27% for Content-moderation, and 56% Verification) while significantly slower in some others (20% slower for AMZ-reviews, 55% for Verification, and 32% Inequality-OA). A similar result can also be observed in the

\(^8\)To approximate the decision time for $P_1$ & $P_2$, we determine the median decision time (per document) for conditions $P_1$ and $P_2$ separately. Then for each document, we use the "slower" predicate as the decision time.
worker answers one of the were between 21% and 55% shorter than the baseline and \( P \). The high dimensionality of the problem makes it intractable to simulations to assess how the performance of workers could vary under different parameterizations of the problem. However, this suggestion demands further research, and we find it

4.2 Simulations

The high dimensionality of the problem makes it intractable to crowdsourcing for every possible parameter value. Here, we rely on simulations to assess how the performance of workers could vary under different parameterizations of the problem.

Conditions. The baseline task asks the complex predicate \( P \) directly, the same-task alternative queries the simpler predicates \( \{p_1, \ldots, p_n\} \) in one task (i.e., a worker answers \( n \) questions), and separate-tasks delivers these predicates on different tasks (i.e., a worker answers one of the \( p_j \) predicates). We use the terms conditions, cases and task alternatives, interchangeably.

Parameters & Metric. We parameterize the simulations based on 1) the number of simpler predicates \( n \) that constitute \( P \), 2) the selectivity \( s_j \) for predicates \( p_j \in P \), 3) the accuracy of workers drawn from a Beta distribution with mean \( \mu \) and variance \( \sigma^2 \), 4) the budget \( b \) controlling the number of votes per item, and 5) the penalty \( y \) that impacts the accuracy of the complex predicate \( P \). To assess different quality goals, we use \( F_2 \), for several values of \( y \).

Worker accuracy. For a complex \( P \), the separate-tasks condition defines a beta distribution for each predicate with expected accuracy \( \mu_j \) for \( p_j \in P \). The same-task condition defines a beta distribution with accuracy \( \mu_k = \frac{1}{n} \sum u_j \). In contrast, the baseline defines a beta distribution with expected accuracy \( \mu_b \) but adjusted based on the penalty \( y \).

We describe results for settings without penalty \( (y = 0) \) and summarize the impact of \( y \) at the end of this section, referring readers to our supplementary materials for further details on our parameterization and in-depth analysis.

Equal selectivity and accuracy. In this scenario, we define that predicates have equal selectivity \( s \), and workers come from the same distribution. Figure 3 depicts the results for \( n = 2, s = 0.5 \), and for different expected accuracy values. It can be noticed than when precision and recall weigh equally \( (\beta = 1) \), there is a difference between the task alternatives in favor of the baseline. However, the gap decreases as the accuracy of workers increases (until the conditions perform roughly the same). The same-task and separate-tasks alternatives outperform the baseline when precision is more relevant than recall \( (\beta = 1) \) and workers are better than random \( (\mu \geq 0.6) \). However, the conditions perform roughly the same when we consider higher selectivity \( (s > 0.5) \). The baseline outperforms the other alternatives when we value more recall \( (\beta = 10) \), and the difference holds as we increase the accuracy and selectivity (except for \( s = 0.1 \), a extremely low selectivity with high variance).

Increasing the budget (number of votes) does not affect the results in low-accuracy settings. But when accuracy is high \( (\mu \geq 0.7) \), the differences narrow until the conditions perform roughly the same. The number of predicates, however, harms the baseline performance, making the same- and separate-tasks superior choices for all settings.

These observations suggest that we may seek to formulate a complex predicate as a single question if we aim to optimize recall and the number of simpler predicates \( n \) is low, which intuitively makes sense and aligns with current guidelines for multi-class classification [17]. While if we aim for precision or face scenarios with many predicates, we are better off by querying a complex predicate via its constituents. However, in the following, we assess more realistic settings (different selectivities and accuracies), and see how asking the simpler questions is preferable over \( P \).

Different selectivity and same accuracy. We assign different selectivity values (either low or high) to the predicates, and we assume the same expected accuracy for the individual predicates \( p_j \). We first considered two predicates \( p_1 \) and \( p_2 \) and two scenarios where the predicates have selectivities 1) \( s_1 = 0.3 \) and \( s_2 = 0.7 \); and 2) \( s_1 = 0.7 \) and \( s_2 = 0.3 \). We tested different accuracies \( \mu \in [0.5, 0.9] \).

| Condition | Exergame-VR | Inequality-OA | AMZ-reviews | Wiki-detox | Verification |
|-----------|-------------|--------------|-------------|------------|-------------|
| Baseline | 0.600 (0.571) | 0.691 (0.476) | 0.909 | 0.697 | 0.674 |
| P1 - P2 | 0.583 (0.696) | 0.781 (0.606) | 0.889 | 0.825 | 0.707 |
| P1 & P2 | 0.656 (0.629) | 0.698 (0.435) | 0.947 | 0.642 | 0.619 |
| P1 | 0.819 (0.817) | 0.744 (0.595) | 0.981 | 0.838 | 0.889 |
| P2 | 0.881 (0.853) | 0.887 (0.942) | 0.926 | 0.719 | 0.776 |

80-20 scenario. This suggests worker strategies such as short-circuit evaluation or focusing on simpler criteria when evaluating complex predicates, but the behavior requires further exploration.

To complement our analysis, we also explore how the predicate formulation may have influenced task intake. Overall, the percentage of workers who quit after a quick inspection of the task (during training) ranged between 18% and 73%. In particular, the task abandonment in the \( P_1\) condition ranged between 50% and 73% (somewhat expected given that workers faced the same amount of instructions as in the baseline and had to answer two questions rather than one). Across all datasets, either \( P_1 \) or \( P_2 \) obtained the highest task intake, aligning with the observation from the previous paragraph. To aid task intake, as task designers, we may seek to formulate a complex predicate as multiple (focused and simpler) questions and query them in isolation. Also, the instructions length should be kept in mind, in the \( P_1 \) and \( P_2 \) conditions our instructions were between 21% and 55% shorter than the baseline and \( P_1\) and \( P_2 \). However, this suggestion demands further research, and we find it an interesting direction to explore.

Our results show that, as current literature suggests, there is a trade-off between quality and time. Besides, formulating complex predicates as multipart questions could also help identify which predicates may be more effort-intensive. Please refer to our supplementary material for a more detailed analysis.
We tested all combinations of selectivity and accuracy. We noticed a difference in performance in favor of the same- and separate-tasks condition also showed superior performance in previous simulations, putting more weight to precision favors separate-tasks conditions (though less pronounced for elementary material). When we weight recall and precision equally, the difference between the conditions when accuracy is high (until the baseline performance, and increasing the budget narrowed the difference). We also considered the case of multiple predicates (n = 4) with different accuracies and selectivities. Like in previous simulations, a higher number of predicates hurts the baseline performance. In this setting, the same- and separate-tasks conditions outperformed the baseline across different values of budget b.

Summary. Our simulations without penalty showed how formulating a composite predicate as a single question is preferable for recall if we consider a small number of predicates with equal selectivity and accuracy. However, this is not always the case in real-world settings, where we have many predicates with different accuracy and selectivity. In these contexts, we noticed that formulating a complex predicate P as multiple simpler questions showed superior performance in general, which aligns with our real-world experiment. As we increase the penalty (γ > 0), the baseline tends towards 0.5 (random guessing), and naturally, the performance deteriorates, making the conditions that ask the individual predicates more suitable.

5 HYBRID CLASSIFICATION

5.1 Problem definition

We extend the crowdsourced classification by allowing to employ a set M of machine learning (ML) classifiers. We want to identify the items in I that meet the complex predicate P, but we now can use ML classifiers alongside crowd workers.

To solve this problem, we now consider training ML classifiers as we cast votes from the crowd W. The classifiers M can be trained for P directly, or for (some of) the simpler predicates pj ∈ P. Therefore, the solution space is naturally impacted by how well the ML classifiers can learn P or the individual constituents, and thus help in the crowdsourced classification problem.

5.2 Experiment

The crowdsourcing experiment showed how performance gains are obtained by querying a complex predicate as multiple (simpler) questions and then combining back the results. Here we situate the predicate formulation problem in the context of hybrid classification and test our insight from the crowdsourcing experiment. The literature suggests that hybrid classification offers superior results. And our intuition is that formulating a complex predicate as multiple questions would allow us to capitalize on the strength of crowd and ML classifiers and, therefore, obtain superior performance.

Design. We consider four datasets from the crowdsourcing experiment (excluding Verification), with 118 items in each dataset. The crowd judgments from the crowdsourcing experiment are aggregated using majority voting, and we combine these with machine predictions in two (simplistic) ways: Crowd-ML and ML-Crowd, where Crowd-ML leverages the crowd for the first predicate (p1) and machine for the second (p2), while ML-Crowd does the opposite (computing p1 ∧ p2 to derive the complex P).

We use classifiers and ensembles of classifiers in this experiment. The four machine learning classifiers correspond to Logistic Regression (LR), Support Vector Machine (SVM), BERT [3] and DistilBERT.
The aim of covering different ML techniques is to give our analysis breadth and not to compare the models, primarily since we are operating with small datasets. The ensemble methods use LR, SVM, and Multinomial Naive Bayes (MNB) as base estimators. We considered voting classifiers (‘hard’, using majority voting, and ‘soft’, using the predicted probabilities), a bagging classifier (with SVM as its base estimator), and a stacking classifier.

The models were trained on the complex \( P \) and its constituents. We used 10-fold stratified cross-validation, repeating the experiment 10 times (with different seeds) and reporting averages. We fine-tuned the deep learning models for 4 epochs with a learning rate of 0.001 using the AdamW optimizer [30]. We used an oversampling technique [18] to aid the LR, SVM, and MNB classifiers (alone and within an ensemble) in dealing with imbalanced classes.

### 5.3 Results

Hybrid classification, Table 2, showed a superior (or comparable) performance when compared to crowd classification for most of the datasets we considered (see our supplementary material for a more in-depth analysis). For the 60-40 case, the hybrid classifier outperformed the crowd for the Exergame-VR and Inequality-OA datasets (16% and 2% difference in performance, respectively). For AMZ-reviews, the performance was comparable (both classifiers with \( F_1 > 0.9 \)) while for Content-moderation the crowd showed superior classification with a score of \( F_1 = 0.82 \) in comparison to only 0.48 for the hybrid approach (this was a difficult dataset in general for both crowd and ML classifiers). The hybrid classification outperformed in the 80-20 variant of the Exergame-VR dataset (9% difference in \( F_1 \)), while the crowd obtained a slightly better performance for the 80-20 version of Inequality-OA (3% difference).

Hybrid classification outperformed ML for AMZ-reviews (21% difference) and Content-moderation datasets, although the hybrid performance was almost random for Content-moderation. In contrast, ML performed better for Exergame-VR and Inequality-OA datasets (11% and 6% difference, respectively), including the imbalanced variants, where the difference was at most 10%.

From a task design perspective, these results suggest that framing a complex predicate as multiple simpler questions translates into performance gains and plays nicely with recommendations from hybrid classification research. Querying a complex predicate \( P \) via its constituents allows for a (potentially) better coupling of crowd and machine classifiers. Our experiment showed that even this simple Human-AI collaboration approach gives a solid and consistent performance across different datasets and domains.

### 6 DISCUSSION & CONCLUSION

Performance gains could be obtained depending on how we formulate a composite question in the context of crowdsourced and hybrid classification. From a task designer perspective, leveraging focused more straightforward questions offers more detailed information about crowd workers, and can inform the use of different approaches more adapted to the characteristics (difficulty, selectivity) of each simpler predicate, instead of committing to a single strategy (e.g., hiring different workers based on task difficulty [25, 43]).

Querying simpler predicates could enable more effective coupling of ML classifiers and favor long term reusability of already trained models. We believe that there is potential for training highly-specialized models that couple effectively with the performance of workers (instead of learning models classify items based on complex predicates directly). Besides, answering simpler questions outputs reusable (and detailed) knowledge about the capabilities of crowd and machine classifiers. For example, if we were to work on an SLR about exergame usage in older adults, we could rely on the current knowledge that we have built by querying the simpler predicates from the Exergame-VR and Inequality-OA datasets. From the perspective of crowd workers, this means reapplying learned skills, and for machines, it involves classifying unseen papers (and filter out at least articles that are “obviously” not relevant).

We focused on a specific but relevant aspect for task designers: how to frame a composite question used to classify items. Our results showed that superior classification performance could be obtained by querying a complex predicate as multiple (simpler) questions instead of asking a single coarse predicate. In a scenario with low accuracy and selectivity, asking the constituents of \( P \) (i.e., \( n \) questions) may increase the chances of misclassifying items, as observed in our simulations. In this case, we may rely on framing the complex \( P \) as a single question (limited by the number of predicates it contains) or framing \( P \) as a mix of simpler and coarse questions. To some extent, our competing results from either asking predicates on the same task vs. on separate tasks is related to this point (i.e., the error rate of a single worker answering \( n \) questions vs. \( n \) workers answering a question each). Both task design choices offer superior results over the baseline, but there is not enough evidence to inform decisions based on given problem settings. We find this an interesting direction of future work, where we design algorithms that model workers, tasks, and predicates to automatically learn how to formulate complex predicates to meet quality goals while operating under a budget.

| Classifier | Exergame-VR 60-40 (80-20) | Inequality-OA 60-40 (80-20) | AMZ-reviews 60-40 | Content-moderation 60-40 |
|------------|---------------------------|-----------------------------|-------------------|-------------------------|
| Crowd      | 0.656 (0.696)             | 0.781 (0.606)               | 0.947             | 0.825                   |
| ML         | **0.866** (0.821)         | **0.853** (0.651)          | 0.753             | 0.183                   |
| Hybrid     | 0.773 (0.762)             | 0.800 (0.588)              | 0.931             | 0.485                   |

Table 2: The cells correspond to \( F_1 \) scores for the best crowd performance (\( P_1 - P_2 \) vs. \( P_1 \) & \( P_2 \)), the best ML result on average (among single and ensemble of classifiers), and the best hybrid performance (Crowd-ML vs. ML-Crowd). The standard deviation for ML is \( \leq 0.07 \).
ACKNOWLEDGMENTS

This work was supported by the Russian Science Foundation (Project No. 19-18-00282).

REFERENCES

[1] Dana Chandler and Adam Kapelner. 2013. Breaking monotonous with meaning: Motivation in crowdsourcing markets. Journal of Economic Behavior & Organization 90 (2013), 123–133.
[2] Justin Cheng and Michael S. Bernstein. 2015. Flock: Hybrid Crowd-Machine Learning Classifiers. In CSCW 2015.
[3] Jacob et al. Devlin. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805 (2018).
[4] Olive Jean Dunn. 1964. Multiple Comparisons Using Rank Sums. Technometrics 6, 3 (1964).
[5] Aditya G. Parameswaran et al. 2012. CrowdScreen: algorithms for filtering data with humans. In SIGMOM 2012.
[6] Ayush Jain et al. 2017. Understanding Workers, Developing Effective Tasks, and Enhancing Marketplace Dynamics: A Study of a Large Crowdsourcing Marketplace. PVLDB 10, 7 (2017).
[7] Angli Liu et al. 2016. Effective Crowd Annotation for Relation Extraction. In NAACL HLT 2016.
[8] Chien-Ju Ho et al. 2015. Incentivizing High Quality Crowdwork. In WWW 2015.
[9] Evgeny Krivosheev et al. 2017. On the impact of predicate complexity in crowdsourced classification tasks. In ICWE 2017.
[10] Chien-Ju Ho et al. 2015. Incentivizing High Quality Crowdwork. In WWW 2015.
[11] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. arXiv preprint arXiv:1901.07367 (2019).
[12] Tanushree Mitra, Clayton J. Hutto, and Eric Gilbert. 2017. Comparing Person- and Process-centric Strategies for Obtaining Quality Data on Amazon Mechanical Turk. In CHI 2015. 1345–1354. https://doi.org/10.1145/2702123.2702553
[13] Michael L. Mortensen et al. 2016. An exploration of crowdsourcing citation screening for systematic reviews. Research Synthesis Methods (2016).
[14] Doren Lan et al. 2017. Dynamic Filter: Adaptive Query Processing with the Crowd. In CHI 2017.
[15] Weizhao Wang et al. 2016. The Impact of Task Abandonment on Task Completion Time in Crowd Platforms. In CSCW 2016.
[16] Donald Green and Stephen E. Sennott. 1974. The robust design of experiments. Technometrics 16, 1 (1974), 1–17.
[17] Justin Cheng and Michael S. Bernstein. 2015. Flock: Hybrid Crowd-Machine Learning Classifiers. In CSCW 2015.
[18] Steven Dow et al. 2012. Shepherding the crowd yields better work. In CHI 2012.
[19] Hyunjung Park and Jennifer Widom. 2013. Query Optimization over Crowdsourced Data. PVLDB 6, 10 (2013), 781–792. https://doi.org/10.14778/2536206.2536207
[20] Nathanial D. Porter, Ashton M. Veredy, and S. Michael Gaddis. 2020. Enhancing big data in the social sciences with crowdsourcing: Data augmentation techniques, practices, and opportunities. PLOS ONE 15, 6 (2020), 1–21. https://doi.org/10.1371/journal.pone.0231514
[21] Rehab Kamal, Qarout, Alessandro Checco, Gianluca Demartini, and Kalina Bontcheva. 2019. Platform-Related Factors in Repetitiveness and Reproducibility of Crowdsourcing Tasks. In IJCAI 2019.
[22] Lei Han et al. 2019. All Those Wasted Hours: On Task Abandonment in Crowd-Sourcing. In IJCAI 2019.
[23] Hyunjung Park and Jennifer Widom. 2013. Query Optimization over Crowdsourced Data. PVLDB 6, 10 (2013), 781–792. https://doi.org/10.14778/2536206.2536207
[24] Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. 2017. Clarity is a Worthwhile Task Crowdsourcing for Complex Data Processing. In HT 2017.
[25] Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. 2014. A taxonomy of microtask Crowdsourcing Classifiers. In CSCW 2015.
[26] Justin Cheng et al. 2015. Break B Down: A Comparison of Macro- and Microtasks. In IJCAI 2015.
[27] Michael J. Franklin et al. 2011. CrowdDB: answering queries with crowdsourcing. In ICDE 2011.
[28] Markus Krause and René F. Kizilcec. 2015. To Play or Not to Play: Interactions between Response Quality and Task Complexity in Games and Paid Crowdsourcing. In CSCW 2020.
[29] Ayush Jain et al. 2017. Understanding Workers, Developing Effective Tasks, and Enhancing Marketplace Dynamics: A Study of a Large Crowdsourcing Marketplace. PVLDB 10, 7 (2017).
[30] Tanushree Mitra, Clayton J. Hutto, and Eric Gilbert. 2017. Comparing Person- and Process-centric Strategies for Obtaining Quality Data on Amazon Mechanical Turk. In CHI 2015. 1345–1354. https://doi.org/10.1145/2702123.2702553