CrowdED and CREX : Towards Easy Crowdsourcing Quality Control Evaluation

Tarek Awwad, Nadia Bennani,
Lionel Brunie
University of Lyon, CNRS, INSA
Lyon, LIRIS - France
first.last@insa-lyon.fr

Harald Kosch
University of Passau
Passau, Germany
harald.kosch@uni-passau.de

Veronika Rehn-Sonigo
FEMTO-ST Institute, Université
Bourgogne Franche-Comté / CNRS
Besançon, France
veronika.sonigo@femto-st.fr

ABSTRACT

Crowdsourcing is a time- and cost-efficient web-based technique for labeling large datasets like those used in Machine Learning. Controlling the output quality in crowdsourcing is an active research domain which has yielded a fair number of methods and approaches. Due to the quantitative and qualitative limitations of the existing evaluation datasets, comparing and evaluating these methods have been very limited. In this paper, we present CrowdED (Crowdsourcing Evaluation Dataset), a rich dataset for evaluating a wide range of quality control methods alongside with CREX (CReate Enrich eXtend), a framework that allows and facilitates the creation of such datasets and guarantees their future-proofing and reusability through customizable extension and enrichment.

KEYWORDS

Crowdsourcing, quality control, dataset, generic platform, extendable campaign

ACM Reference Format:
Tarek Awwad, Nadia Bennani, Lionel Brunie, Harald Kosch, and Veronika Rehn-Sonigo. 2019. CrowdED and CREX : Towards Easy Crowdsourcing Quality Control Evaluation. In *Proceedings of* . ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the era where Artificial Intelligence is emerging at a steady fast pace through its underlying concepts such as Machine Learning and Data Mining, the quest for collecting labeled data like labeled images or annotated metadata is a persistent and fundamental task for researchers in these domains. In the last decade, crowdsourcing has proved its ability to address this challenge by providing a mean to collect labeled data of various types, at a low cost and short time as compared to expert labeling. However, the quality of the data produced through crowdsourcing is still questionable, especially when the labeling task shows a fair amount of subjectivity or ambiguity or requires some domain expertise [38].

Tackling this quality issue is, consequently, an active research domain that has yielded a large number of quality control methods ranging from optimizing the contribution aggregation process [12, 16, 41] and the worker selection step [6, 26] to modeling context-specific reputation systems [28, 29] and controlled crowdsourcing workflows [13]. Indeed, validating and comparing these methods raise the need for evaluation datasets which are sufficiently representative, information rich and easily extensible. Existing datasets [11, 19, 39, 40, 44] do not fulfill those requirements because they are tailored, form-wise, to evaluate one method or in the best cases, one category of approaches. This renders the cross-category comparison - like comparing aggregation approaches to selection approaches - unfeasible through sound scientific workflows. To address this challenge we designed and collected CrowdED (Crowdsourcing Evaluation Dataset), a publicly available information-rich evaluation dataset. In this paper, we detail and motivate the creation of CrowdED and describe CREX (CReate Enrich eXtend), an open platform that allows and facilitates the collaborative extension and enrichment of CrowdED. The contributions of this paper can be summarized as follows :

- We provide a comprehensive specification sheet for a generic and future proof evaluation dataset, provide a comparative review of the existing datasets and discuss their compliance with those specifications.
- We propose CrowdED, a rich evaluation dataset of which we present the design and the contribution collection steps as well as the statistical and structural properties.
- We assess the ability of CrowdED in plugging the dataset gap through a qualitative study.
- We present the design of CREX and show how it facilitates the creation of crowdsourcing campaigns to extend and enrich evaluation datasets similar to CrowdED.

This paper is structured as follows: In Section 2, the state of the art of quality control methods is briefly reviewed. In Section 3, the specifications of a suitable evaluation dataset are set. In Section 4, the state of the art crowdsourcing evaluation datasets are discussed w.r.t. the requirements stated earlier. Then, in Section 5, we describe the creation process of CrowdED as well as its structural and statistical characteristics. Finally, we present CREX in Section 6 and discuss its reusability in Section 7, before concluding this paper in Section 8.

2 CROWDSOURCING QUALITY CONTROL

Many methods have been proposed to perform quality control in crowdsourcing systems [6, 8, 17, 25, 26, 28, 32, 36]. Most works in

, ,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

this domain have focused on optimizing the contribution aggregation process which consists in inferring the correct answer of a task using the collected contributions for this task. Early works used majority voting (MV) with multiple assignments to infer the correct answer to a given task. Giving different weights to the different votes improves the quality of the aggregation by penalizing less reliable answers. Those weights can be computed as graded and binary accuracies [17], credibility scores [28] or overall approval rates which are widely used in commercial crowdsourcing platforms e.g., Figure-Eight and AMT. More generic and widely used techniques [7, 26, 41] rely on probabilistic data completion methods like the expectation maximization algorithm (EM) [9, 10]. In the latter, the weights and the correct answers are simultaneously inferred by maximizing a likelihood model. Li et al. [26] use, in their model, the worker accuracy and inaccuracy as weights for correct and wrong answers (respectively), while in [41], a Generative model of Labels, Abilities, and Difficulties (GLAD) is proposed; GLAD uses both the worker ability and the task difficulty as weights for the contributions in the aggregation process. In [36], the worker's reliability score is estimated using her participation behavior e.g., time for completing a task, the number of clicks, mouse travel, etc. Some methods propose to add more knowledge to the aggregation process using multiple stage crowdsourcing such as the produce/review workflow described in [7].

Another way of controlling the quality consists in allowing only reliable workers to participate to the task completion. This can be done through pre-assignment qualification tests. Platforms like Figure-Eight use a gold-based quality assurance [22] which consists in continuously measuring the accuracy of the worker, using test tasks - with known answers - randomly injected in the workflow. A high error rate causes the rejection of the worker from the current campaign. Programmatic gold [30] is an extension of the gold-based quality control where test tasks with incorrect answers are also used to train workers against common errors. Li et al. [26] propose a probing-based selection method. They describe an algorithm that finds, for each incoming task, a group of reliable online workers for this particular task. This is done by assigning, during the so called *probing stage*, a part of the tasks to the whole crowd in order to sample it and identify the reliable group for the remaining part. Awwad et al. [6] substitute the probing stage by an offline learning phase to learn the reliable group from previously completed tasks with a lower cost. Roy et al. [35] characterize in the same feature space the tasks by the skills they require and the workers with their skills, and then match workers and tasks according to their skills.

Moreover, some approaches in the literature leverage the worker incentive and preference aspects of the crowdsourcing process. For instance, in [4, 5], the authors argue that proposing a personalized (based on the preferences) list of tasks for a given worker improves her throughput in terms of quality. Kamar et al. [18] propose incentive mechanisms that promote truthful reporting in crowdsourcing and discourage manipulation by workers and task owners.

3 SPECIFICATIONS

In this section, we analyze the requirements of the aforementioned quality control approaches and deduce four specifications of a suitable evaluation dataset.

Table 1: The needs of selected quality control methods in terms of dataset content

Method	Workers	Task	Contrib.	Other
MV, WMV [17], EM [9]	ID	ID	Yes	n/a
Worker modeling [20]	ID	ID	Yes	n/a
Task modeling [41]	ID	ID	Yes	n/a
Qualification tests, test questions *	ID	ID/Content	Yes	n/a
History-based *	ID	ID	Yes	n/a
Contextual history-based *	ID	Content	Yes	n/a
Programatic gold [30]	ID	ID/Content	Yes	Online interaction
Profile-based [21, 26]	Profile	ID	Yes	n/a
Contextual profile-based [6]	Profile	Content	Yes	n/a
Skill-based [35]	Skill Profile	Skill Profile	Yes	n/a
Self-evaluation [16]	Profile	Content	Yes	n/a
Task composition [4]	Preferences	Content	Yes	n/a
Incentive based [27]	ID	ID	Yes	Reward variance

n/a: not available, * : Common methods in commercial crowdsourcing platforms.

Specification 1 : Data richness (S1)

Table 1 summarizes the requirements of a representative set of quality control methods. The majority of classical quality control methods such as aggregation techniques [9, 17] do not require any specific features to be present in the dataset aside from the set of contributions, i.e., a set of labels indexed by (ID_{worker}, ID_{task}) keys. Those are indeed required by all the existing methods. Other methods, such as profile-based worker selection [6, 16, 21, 26] necessitate the presence of the worker profiles[1] in the dataset. Methods which take into account the type of the task when selecting/screening workers - and which we refer to as contextual methods - necessitate either the existence of a category-labeled task or the content of the task from which the task type can be derived [35]. Finally, some methods [6] can require information on both the workers and the tasks to be present in the dataset at the same time. Based on this description, we distinguish two specifications related to the richness of a suitable evaluation dataset:

- **S1.1** The dataset must provide information about workers, that is, the worker declarative profiles.
- **S1.2** The dataset must provide information about tasks, that is, their full content, i.e., description, questions and answer options.

Specification 2: Data diversity (S2)

Crowdsourcing tasks cover a wide range of types [38]. Similarly, workers in a crowdsourcing system fall into multiple profile groups [15]. In order to allow assessing the genericity of the compared methods, it is crucial that the evaluation dataset reflects - to a sufficient extent - this type and profile diversity. Accordingly, we set two specifications related to the data diversity :

- **S2.1** The dataset must reflect the diversity of the profile features characterizing the workers of a real crowdsourcing system.
- **S2.2** The dataset must reflect the diversity of the task types. This includes the generic asked action e.g. labeling an image,

[1] E.g., demographics and self-evaluation profiles

judging relevance, analyzing sentiment in text, etc., and the actual knowledge domain of the task e.g. sport, economy, botany, etc.

Specification **S.2** tightens the Specification **S.1**. Indeed, an information cannot be diversified (**S.2**) without existing in the dataset in the first place (**S.1**). Hence, it is possible to drop the looser specification **S.1** while maintaining the completeness of the requirement sheet. Since the opposite is not necessarily true, i.e., an information might exist without being diversified, we keep both specifications to allow a more fine-grained comparison of existing datasets.

Specification 3: Contribution abundance (S3)

To control the quality, one might need to estimate the global [43] or the contextual [6] reliability of the worker from his previous or current contributions, to compute the difficulty of the task using the workers' agreement on its answer [41], to assess the accuracy and the convergence ability of a proposed aggregation method [17, 39], to compute the correlation between worker's reliability (computed using his contributions) and his declarative profile [26] etc. All this requires the dataset to provide sufficient contributions per worker and per task while ensuring that these contributions provide information about the worker reliability and the task difficulty.

We formulate the previous by the following specifications:

S3.1 The dataset must contain a large number of contributions. That is, both the tasks and the workers present in the dataset must have a reasonable number of contributions.

S3.2 The dataset must contain non-random contributions for tasks and for workers. We show later how this can be achieved during the campaign design and the data preprocessing steps.

Specification 4: Extensibility (S4)

The creation of a generic and information rich dataset should always be open to new contributors, so that absent and new features can be proposed and collected based on uncovered and new quality control needs. Moreover, creating a realistic evaluation dataset for crowdsourcing quality control necessarily passes by a crowdsourced data collection step, which is obviously a paid process. This makes the creation of a large enough dataset very costly, hence not achievable by only one entity (research laboratory, company, ...). Therefore, we add to the qualitative specifications **S.1**, **S.2** and **S.3** detailed earlier in this section a fourth specification as follows:

S4.1 The dataset must be collaboratively extensible both in terms of tasks, workers and contributions and in terms of worker features and task types.

In the remainder of this paper, we show how we design and build CrowdED to fulfill Specifications **S.1**, **S.2** and **S.3** and how CREX guarantees its extensibility to fulfill Specification **S.4**.

4 STATE-OF-THE-ART OF CROWDSOURCING EVALUATION DATASETS

Table 2 details the characteristics of the evaluation datasets available in the crowdsourcing literature. For the sake of completeness, both publicly available and non-publicly available datasets are reported even though the latter ones are not accessible and thus cannot be used as benchmarking dataset. The table also shows the compliance of these datasets with Specifications **S.1**, **S.2** and **S.3**. As none of these datasets is compliant with S4, this specification is not shown in the table. The compliance with those specifications is judged based on a set of observed characteristics in the dataset which we enumerate as follows :

- The worker features (*Feat.*) : is the number of worker profile features found in the dataset (related to **S.1.1** and **S.2.1**).
- The task content (*Cont.*) : shows whether the dataset contains information about task content or not (related to **S.1.2**).
- The task diversity (*Div.*) : shows whether the dataset contains more than one type of tasks or not (related to **S.2.2**).
- The contribution density (*Den.*) : shows whether the set of contributions is Dense (D), i.e., all of the tasks were solved by all of the workers, Semi-Dense (DS) , i.e., the sets of workers who answered different tasks overlap or Sparse (S), i.e., the workers who answered one task are different from those who answered another task (related to **S.3.1**).

In the literature, many datasets have been used to evaluate crowdsourcing quality control techniques. Only a few among those provide information about the declarative profile of the workers [16, 26] which is in line with the low number of quality control methods leveraging this aspect. The same observation was made by Ye et al. in [43]. The previous reasoning also applies on the content of the tasks which is not always present in the datasets [16, 39]. On the opposite side, the contribution abundance requirement is almost met by all of the datasets [11, 16, 19, 22, 26, 39, 40]. This might be due to the fact that aggregation methods, which constitute a large part of the crowdsourcing related literature as shown in Section 2, usually require this requirement to be met.

The *Data For Everyone (DFE)*[2] corpus from Figure-Eight provides a large number of real task sets for which many contributions have been collected. While these sets are varied enough in the task types, they suffer from at least one of the following limitations: First, the majority of them provide aggregated contributions instead of individual contributions, which violate Specification **S 2.1**. Second, to the best of our knowledge, none of these datasets provide the profiles of the workers which violates Specification **S 1.1**. Third, the content of the task is not always present which does not meet Specification **S 1.2**. One can argue that it is possible, through some data engineering effort, like transferring missing data like profiles from one set to the other, to combine a number of these sets into a larger specification-fulfilling dataset. However, the datasets found in the DFE corpus are designed and generated independently by different requesters. Hence, the intersection between the workers and tasks of different datasets, when computable e.g., for unaggregated or un-anonymized datasets, might be empty or sparse which hinders any "match and transfer" step.

The aforementioned datasets are all *real crowdsourcing datasets*. That is, datasets generated through an actual crowdsourced data collection step. Alternatively, *Synthetic datasets* have been also used in the literature. Roy et al. [35] and Rahman et .al [31] generated a set of workers and tasks distributed over a set of skills found in a multilayer skill taxonomy in order to test the efficiency of their skill matching approaches. Others, such as Welinder et al. [40] and

[2] https://www.figure-eight.com/data-for-everyone/

Table 2: A comparison of a sample of dataset used in the literature to evaluate crowdsourcing quality control.

		Characteristics									Compliance with our requirements					
Ref	Dataset	Worker		Tasks			Contributions		Pub.	RD	S1.1	S1.2	S2.1	S2.2	S3.1	S3.2
		#	Feat.	#	Cont.	Div.	#	Den.								
[16]	Stack overflow	505	8	14021	Yes	No	42063	S	-	+	+	+	+	-	+	+
	Evergreen webpage	434	9	7,336	Yes	No	22,008	S	-	+	+	+	+	-	+	+
	TREC 2011	160	9	1826	Yes	No	5478	S	-	+	+	+	-	-	+	+
[22]	Online product search	255	0	256	No	No	NA	S	-	+	-	-	-	-	NA	+
[24]	Synthetic	11	0	300	No	No	3300	D	-	-	-	-	-	-	-	-
[26]	Knowledge dataset	100	5	75	Yes	No	7500	D	-	+	+	+	-	-	-	+
	RTE	NA	5	80	Yes	No	NA	D	-	+	+	+	-	-	-	+
	Disambiguation data	277	5	50	Yes	No	13850	D	-	+	+	+	+	-	-	+
[39]	Affective text analysis	10	0	700	Yes	No	7000	D	+	+	-	+	-	-	-	+
	RTE	10	0	800	Yes	No	8000	D	+	+	-	+	-	-	-	+
	Word Similarity	10	0	30	Yes	No	3000	D	+	+	-	+	-	-	-	+
[40]	Image annotation Synth.	4-20 /task	0	500	No	No	NA	NA	-	-	-	-	-	-	-	-
	Image annotation Real	40 /task	0	100	No	No	4000	NA	-	+	-	-	-	-	+	+
[44]	Image labeling	109	0	807	No	No	NA	SD	-	+	-	-	-	-	-	+
	Relevance judgment	6 /task	0	2665	No	No	16000	S	-	+	-	-	-	-	-	+

Feat. : worker Features, **Cont.** : task Content, **Div.** : task Diversity, **Den.** : contribution Density
D : Dense contributions, **DS** : Semi-Dense contributions, **S** : Sparse contributions, **n/a** : not available
Pub. : Public availability, **RD**: Real Dataset, **-** : Un-fulfilled requirement, **+** : Fulfilled requirement

Hung et al. [14], generated synthetic datasets to evaluate the performance of their aggregation algorithms. While generating synthetic evaluation datasets for aggregation and skill matching optimization approaches is relatively an easy and scientifically valid approach, generating synthetic datasets to evaluate approaches that leverage worker's behavior (e.g., fingerprinting [36]) and profile (e.g., declarative profile based worker selection [6, 26]) is unfeasible. That is because, on the one hand, ignoring the uncertainty and noise resulting from the subjectivity of the human being in generating the data, produces a dataset which does not reflect the real crowdsourcing context. And, on the other hand, modeling the uncertainty and noise is impossible due to the lack of behavioral studies of the crowd in crowdsourcing systems. Hence, a synthetic dataset could, theoretically, fulfill all the specifications except Specification **S 3.2**.

To summarize, existing real datasets as well as synthetic data generation are not satisfactory to solve the challenge of evaluating and comparing quality control methods in crowdsourcing.

5 CROWDED : CROWDSOURCING EVALUATION DATASET

In this section, the process used to create CrowdED is described in detail and its statistical and structural characteristics are presented. This process is divided into three steps : First, the data preparation during which the raw resources such as the task input are collected and preprocessed. Second, the data collection step during which the actual contributions and profiles crowdsourcing occurred. Finally, the data formatting step during which the collected contributions and profiles are cleaned and restructured.

5.1 Raw Data Preparation

We built our task corpus by collecting publicly available task sets from the Data For Everyone datasets provided freely by Figure Eight[3] (FE). The main motivation behind choosing the DFE datasets is to use tasks that have served real world applications. In fact, it is possible to generate random labeling and knowledge related tasks from scratch and to use them in the dataset generation process. However, those will not be as significant as real world tasks. Furthermore, DFE is a sustainable source[4] of task sets for future extension of CrowdED (Specification **S 4.1**). Our initial task pool consisted of 280K+ tasks, originally belonging to 11 different task sets. The task content was distributed over various domains such as sport, fashion, politics, economy, disaster relief etc. and over different action types like relevance judgment, image labeling, tweet categorization etc. (Specification **S 2.2**). Table 3 summarizes the characteristics of the task sets used to build the task corpus. It is clear that the tasks are unevenly distributed over the various task sets. For instance, set "A9" constitutes 67% of the entire corpus. That is why, in order to balance our task corpus we sampled 4000 tasks out of each set (i.e., the size of the smallest set). The set of 44k resulting tasks constitutes our task corpus. In the next step, a random sample of 525 tasks[5] within the task corpus was published for crowdsourcing.

Table 3: On overview of the task sets used to build initial task corpus of CrowdED.

Task set name	# Tasks	# Questions	Question type	Domain
A8	18129	3	MCQ	Disaster relief
A9	189000	2	MCQ/FT	Sport
BI	10672	2	MCQ	Natural sciences
CH	5702	1	MCQ	Natural sciences
DE	15702	1	MCQ	Fashion
FO	4000	1	MCQ	Sport
GO	13872	3	MCQ	Politics
PO	5000	3	MCQ	Politics
SO	10976	1	MCQ	Disaster relief
SM	4000	2	MCQ/FT	Technology
US	5015	1	MCQ/FT	Economy

MCQ: Multiple Choice Questions.
FT : Free Text answer questions.

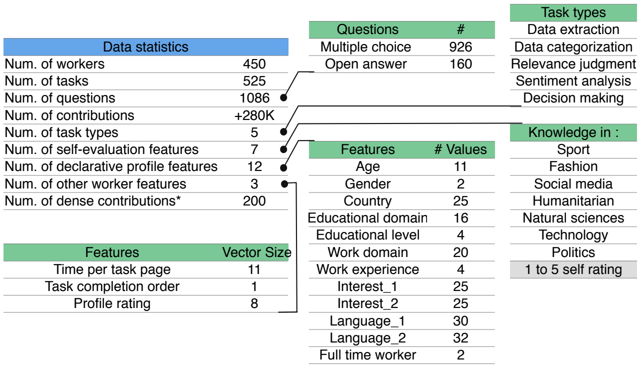
5.2 Data collection

We designed a crowdsourcing job and submitted it to FE. Workers who selected the job were asked to read a detailed description of the task solving process and conditions and to fill their contributor IDs.

[3] https://www.figure-eight.com. Formerly named CrowdFlower.

[4] Yet, it is not the only one since any other task corpus can be used.

[5] Limited by our crowdsourcing budget.



Data statistics	
Num. of workers	450
Num. of tasks	525
Num. of questions	1086
Num. of contributions	+280K
Num. of task types	5
Num. of self-evaluation features	7
Num. of declarative profile features	12
Num. of other worker features	3
Num. of dense contributions*	200

Features	Vector Size
Time per task page	11
Task completion order	1
Profile rating	8

Questions	#
Multiple choice	926
Open answer	160

Features	# Values
Age	11
Gender	2
Country	25
Educational domain	16
Educational level	4
Work domain	20
Work experience	4
Interest_1	25
Interest_2	25
Language_1	30
Language_2	32
Full time worker	2

Task types
Data extraction
Data categorization
Relevance judgment
Sentiment analysis
Decision making

Knowledge in :
Sport
Fashion
Social media
Humanitarian
Natural sciences
Technology
Politics
1 to 5 self rating

Figure 1: An overview of the structural characteristics of CrowdED. (*) a dense contribution is a set of answers given by a single worker to the entire task set.

Those who decided to proceed with the job completion were redirected to an external web page on which the data collection took place. In the first stage of the task, we asked workers to fill their contributor IDs again (for an easier matching and control) and to answer a set of profile related and self-evaluation questions (Specification **S 1.2**)(see Section 5.3). Once done, workers proceeded in the actual task solving. For each job instance, tasks were randomly distributed over 11 pages in order to prevent the concentration of the negative impact of weariness on one subset of tasks. After completing the whole task set, a unique submission code was provided to each worker allowing her to receive her reward on FE.

Workers were rewarded a base pay equal to 1$ US. Additionally, a bonus of 2$ US was awarded (manually) to workers whose answers and declared profiles were of a good quality and high consistency (see Profile Rating in Section 5.3). Moreover, we estimated the job completion time by 45 minutes, thus workers who finished the job in a very short time (i.e., less than 40 minutes) were automatically eliminated and did not receive any reward. Finally, we only accepted workers of at least level 2 in the FE worker classification[6]. on the one hand, these three parameters i.e., the bonus, the contribution duration and the minimum worker level, were strict enough to ensure that malicious workers (i.e., workers who intentionally fill random or wrong answers) are eliminated (Specification S 3.2). On the other hand, they are loose enough to allow a real representation of the quality issue in crowdsourcing. Contributions were collected during 3 months over all week days and covering all times of the day. This is to eliminate the bias related to the time zones, holidays and working hours during the data collection, e.g., workers representing few countries, limited educational and work profiles, etc.

5.3 Data Structure and statistics

Figure 1 shows the structural characteristics of CrowdED as well as the features of tasks and workers that it contains. In total, we collected 280K+ contributions for 525 tasks from 450 workers among which 200 completed the entire set of tasks. We call the set of contributions given by those 200 workers a "dense set". Structurally, CrowdED consists of 4 files: *contributions.csv* which contains the worker contributions, *workers.csv* which contains the worker profiles, *rating.csv* where profile ratings are stored and finally task.zip where the tasks content and description are stored in JSON format. CrowdED have been made public on Figshare and on Github.

5.3.1 Tasks. Some of the 525 tasks in CrowdED contain up to three independent questions. The total number of answered questions is 1086. The majority of these questions (926) are multiple choice questions and the remaining part consists of open answer questions. The input of the tasks are tweets, images, scientific article quotes or news articles and headlines. Their action types fall into five categories: *data extraction, data categorization, relevance judgment, sentiment analysis, and decision making.*

5.3.2 Workers. For each worker, we collected a profile consisting of 21 features divided into three categories:

Declarative profile. We collected 12 features consisting of the following demographical, education and interest related information about the user : *age, gender, country, education domain, education level, work domain, work experience, interests (two features), native language, other spoken language and full time worker (i.e., whether the worker is a full time or occasional crowdsourcing worker).* We observed that these numbers are, for their majority, compliant with the numbers reported in previous studies found in the literature such as the study of the Mechanical Turk marketplace [15].

Self-evaluation features. We collected 7 features consisting of a 5-star self rating for 7 knowledge domains: *sport, fashion, technology, natural sciences, humanitarian work, politics, and social media.* We observed that in average, female workers seemed more confident in their knowledge in fashion and Humanitarian work, while male workers, rated themselves higher for sport. For the remaining domains, i.e., technology, natural sciences, politics and social media, both female and male workers rated themselves similarly.

Behavior-related features. Four features related to the behavior of the workers during the campaign were collected. Three of these features were collected automatically in the interface : *time for completing a task page, time for reading the description and filling the profile and the order of task completion.* The fourth, however, resulted from a complementary crowdsourcing campaign; in fact, in order to judge the consistency and reliability of the worker declarative and self-evaluation profiles, we ran a profile rating job on FE during which the profile of each worker who participated to our job was rated (from 1 to 4) for consistency by at least 7 workers (with an average of 11 workers).

6 CREX: CREATE, ENRICH, EXTEND

Generating the data described earlier is a technically tedious and time consuming task. In this section, we present CREX (CReate, Enrich, eXtend), a framework that allows and facilitates the generation of such data[7]. CREX uses a two-component architecture. This architecture is shown in Figure 2. The first component, CREX-D,

[6] FE levels range from 1 to 3 where level 3 represents the most experienced and reliable workers and 1 represents all qualified workers

[7] Note that CREX has been used to create CrowdED.

allows a configurable task data selection while the second, CREX-C, provides tools to automatically generate crowdsourcing campaigns from the output of CREX-D. The computational modules of CREX are developed with Python3. CREX uses well established and sustainable natural language processing and machine learning libraries such as *scikit-learn*[3], *nltk*[2], *gensim*[1] etc. The web user interface uses a combination of *Bootstrap*, *JavaScript* and *PHP* and the used database technology is *MySQL*[8].

6.1 Data preparation component (CREX-D)

A typical crowdsourcing workflow consists of 3 steps: first, designing the task, second, crowdsourcing the task and last, collecting the results. Indeed, this typical workflow is suitable for classical crowdsourcing where the aim of the requester is to exploit the results in a limited application-centric way, e.g., label multimedia data to facilitate indexing, translate a given corpus, etc. In other words, it suits applications where the input data are fixed and limited in size. When it comes to research-related crowdsourcing, e.g., building evaluation, validation or training datasets where the usage of the collected data goes beyond the limited exploitation, the input data space is usually huge and more complex. Therefore, an upstream input data selection effort is needed. A more suitable workflow is then a four step process that adds an input data selection step at the beginning of the aforementioned workflow. We propose a data selection step encapsulated in the data preparation component CREX-D that allows the requester to group his tasks according to their types through clustering, and then, to reduce their number according to his budget through sampling.

Figure 2 depicts the structure of the CREX-D component. It comprises four modules: the *vectorizing module (CREX-VM)* , the *clustering module (CREX-CM)*, the *sampling module (CREX-SM)* as well as the *evaluation module (CREX-EM)*. Those modules are available and **inter-operable** yet **independent**. That is, each module can be used separately or as an entry point for the remaining steps or substituted by another module of equivalent role. This allows a more flexible usage and thus a wider cross-domains utility of CREX.

The vectorizing module: Grouping the tasks starts by extracting the features of interest from the raw data. In this work, we consider textual data where each data point is the textual representation of a task. Despite being limited to this type of data, CREX makes it easy to bypass this limitation by either feeding pre-vectorized data to the *CREX-CM* or by adding custom vectorizing functions to the *CREX-VM*. The actual implementation of *CREX-VM* supports frequency based text representation (TF-IDF [37]) and semantic document representation (Doc2vec [23]).

The clustering module: The *CREX-CM* allows to cluster the vectorized tasks using one of three types of clustering algorithms: partitional (K-means), density-based (DBSCAN), and hierarchical (Agglomerative). User can natively use either a cosine or an Euclidean distance during the clustering process. However, the CREX-CM provides the possibility to feed the algorithm with a custom pre-computed similarity matrix.

The sampling module: This module allows to sample from an input task corpus a smaller set of tasks that can be crowdsourced while respecting the budget constraints of the requester. This module implements a basic stratified sampling algorithm and a type-aware constrained sampling process which is out of the scope of this paper.

The evaluation module: The *CREX-EM* module allows to evaluate the clustering process using internal and external validity measures such as silhouette [34], homogeneity, completeness and V-measure [33] as well as a custom validity measure consisting of a similarity to co-occurrence correlation matrix.

6.2 Campaign management component (CREX-C)

From a requester perspective, a mandatory step of the crowdsourcing workflow is the task design and generation. This step is tedious and time consuming due to two factors: first, the interest and use of crowdsourcing is growing to reach a wider sphere of scientific and social domains. Thus, the range of task forms and content is getting larger. Second, a crowdsourcing task, itself, might be dynamic, i.e., it may require conditional or real-time computed components. Therefore, it becomes harder for commercial crowdsourcing platforms to quickly adapt their design tools, preset templates and real-time computational means[9]. A common way of dealing with these limitations is to build campaign sites with dedicated databases and back-end computations and to make them accessible through a common crowdsourcing platform to provide reward payment and worker management (for security and trust). The campaign management component of CREX, CREX-C, provides an easy-to-use tool for generating campaign sites from the sampled tasks (see Section 6.1) using the *campaign generator module* (CG).

The campaign generator module: CREX-CG takes two inputs: the set of tasks to be published on the campaign site and the requester input consisting of the task descriptions, examples and instructions. It parses these inputs to intermediate JSON files and uses them to generate the campaign pages. The campaign site communicates directly with the database where the contributions and the worker profiles are stored. Contributions in the database are stored using a JSON format which allows a straightforward use of CREX-C for different task structures and types without the need for a new database model and query rewriting.

The filtering module: For a set of workers, tasks and contributions collected after publishing the campaign generated by the CREX-CG module, the filtering module allows to select a subset of these data based on qualitative and quantitative selection criteria applied on the workers. Those criteria cover the declarative profile features of the worker, their rating of their profiles, their time of task completion, their time of profile completion as well as the number of task they achieved. The filtering process has two main goals: First, it helps selecting a subset of the workers based on qualitative criteria to allow studying its characteristics e.g., its average performance of female workers. Second, it allows to clean the data based on behavioral criteria. For instance, a profile filled in less than 20 second is

[8] A demo of CREX's user interface and a real world use scenario can be found on https://project-crowd.eu/

[9] e.g., requester accessible back-end services or API to **dynamically** modify tasks and assignments.

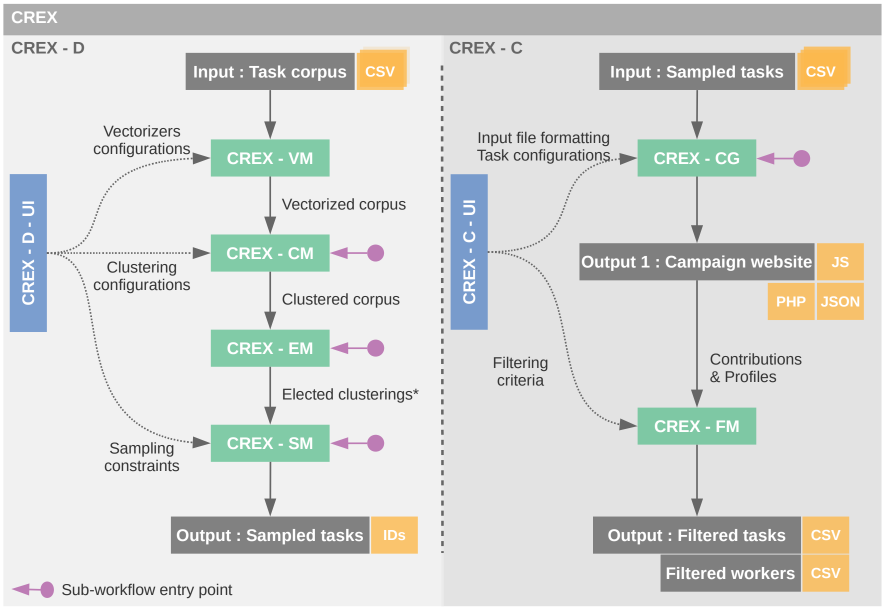


Figure 2: An overview of the CREX framework that combines two main component; CREX-D for data selection and CREX-C for campaign generation and data collection.

most likely to be inconsistent. That is, it has been most likely filled randomly, which means that the worker associated to this profile is very likely a malicious worker. Consequently, considering only the contributions of workers who spent a reasonable time answering the profile questionnaire would yield a noiseless dataset.

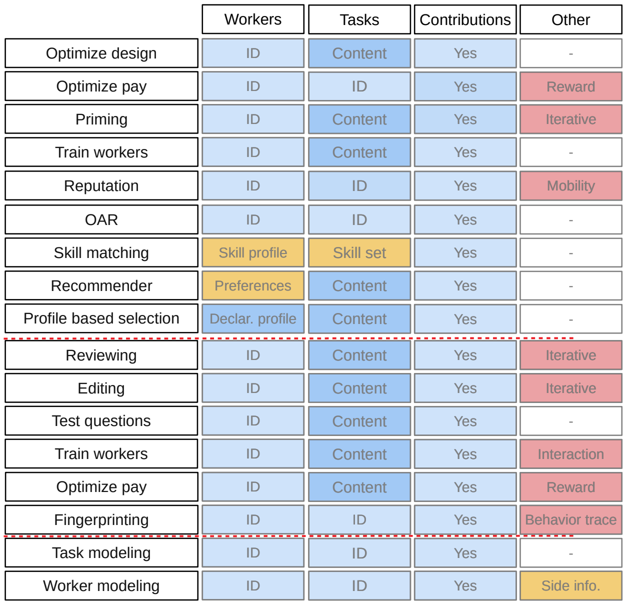
7 CROWDED AND CREX RE-USABILITY

7.1 Usability in quality control evaluation

Table 3 shows the usability of CrowdED for evaluating the quality control methods reported in Table 1. This usability is judged based on the needs of these methods in terms of information about workers, tasks and contributions and their availability in CrowdED. The majority of the methods that require information about workers and tasks only (regardless the type of the information) are natively supported by CrowdED. Others are supported either through *simulation*, i.e., vertically or horizontally splitting the dataset to simulate a real world situation like worker screening or through *augmentation*, i.e., adding more knowledge to the available data without the need for additional crowdsourcing by extracting new features or using external taxonomy to represent tasks and workers. Less frequent methods that require more information are not supported natively. Nevertheless, thanks to CREX, they could be supported by extending CrowdED with a minor reconfiguration effort (e.g. changing the reward) or with a more demanding coding effort.

7.2 Compliance with the FAIR principles

To guarantee the re-usability of those resources by the wide community (which allows a better extension and enrichment of CrowdED), the FAIR principles [42] (*Findable, Accessible, Interoperable, Reusable*) were considered during the design, the creation and the publishing process: CrowdED and CREX are available on Github and Figshare (with an associated DOI) which makes them **F**indable. They are



	Workers	Tasks	Contributions	Other
Optimize design	ID	Content	Yes	-
Optimize pay	ID	ID	Yes	Reward
Priming	ID	Content	Yes	Iterative
Train workers	ID	Content	Yes	-
Reputation	ID	ID	Yes	Mobility
OAR	ID	ID	Yes	-
Skill matching	Skill profile	Skill set	Yes	-
Recommender	Preferences	Content	Yes	-
Profile based selection	Declar. profile	Content	Yes	-
Reviewing	ID	Content	Yes	Iterative
Editing	ID	Content	Yes	Iterative
Test questions	ID	Content	Yes	-
Train workers	ID	Content	Yes	Interaction
Optimize pay	ID	Content	Yes	Reward
Fingerprinting	ID	ID	Yes	Behavior trace
Task modeling	ID	ID	Yes	-
Worker modeling	ID	ID	Yes	Side info.

Figure 3: CrowdED's usability for the existing quality control methods: native, simulation/augmentation, extension.

published under CC and GPL licensing respectively to allow their **R**e-usablility and **A**ccessibility. CrowdED data are stored in *csv* files and no proprietary languages were used to develop CREX. This ensures the **I**nteroperability of the resources.

7.3 Accessibility

The site http://project-crowd.eu/ provides a demo of CREX-D and CREX-C, a tutorial on installing and using CREX, a full description of the configurable parameters as well as additional materials for this paper such as the full statistical sheet of tasks, profiles, ratings and contributions of CrowdED. Moreover, the site provides links to download both CREX and CrowdED.

8 SUMMARY

In this paper we proposed CrowdED and CREX in order to address the lack of evaluation dataset, which is unanimously one of the most challenging aspects facing the research in crowdsourcing quality control. The specifications **S1**, **S2** and **S3** fulfilled by CrowdED allow it to be usable in evaluating and comparing a wide range of existing methods. In order to deal with the methods which are not natively supported by CrowdED, and to future-proof it, we proposed CREX. CREX is an open-source framework that allows the extension of CrowdED to fulfill new qualitative requirements e.g., new worker profile types, and quantitative requirements e.g., more contributions for a given task (**S4**). For the future we plan to extend CrowdED with additional features such as the reward variation and we plan to add other clustering and vectorizing techniques to CREX.

REFERENCES

[1] 2018. Gensim. https://radimrehurek.com/gensim/. (2018). Accessed: 2018-02-25.

[2] 2018. NLTK. https://www.nltk.org/. (2018). Accessed: 2018-02-25.

[3] 2018. Scikit. http://scikit-learn.org/stable/. (2018). Accessed: 2018-02-25.

[4] Maha Alsayasneh, Sihem Amer-Yahia, Eric Gaussier, Vincent Leroy, Julien Pilourdault, Ria Mae Borromeo, Motomichi Toyama, and Jean-Michel Renders. 2018. Personalized and Diverse Task Composition in Crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering* 30, 1 (2018), 128–141.

[5] Sihem Amer-Yahia, Eric Gaussier, Vincent Leroy, Julien Pilourdault, Ria Mae Borromeo, and Motomichi Toyama. 2016. Task composition in crowdsourcing. (2016), 194–203 pages.

[6] T. Awwad, N. Bennani, K. Ziegler, V. Sonigo, L. Brunie, and H. Kosch. 2017. Efficient Worker Selection Through History-Based Learning in Crowdsourcing. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. 923–928. https://doi.org/10.1109/COMPSAC.2017.275

[7] Yukino Baba and Hisashi Kashima. 2013. Statistical Quality Estimation for General Crowdsourcing Tasks. In *ACM SIGKDD*. NY, USA, 554–562.

[8] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques, and Assurance Actions. *ACM Computing Surveys (CSUR)* 51, 1 (2018), 7.

[9] A. P. Dawid and A. M. Skene. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1 (1979), pp. 20–28.

[10] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), 1–38.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. (2009), 248–255.

[12] Arpita Ghosh, Satyen Kale, and Preston McAfee. 2011. Who moderates the moderators?: crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM conference on Electronic commerce*. ACM, 167–176.

[13] Yolanda Gil, Daniel Garijo, Varun Ratnakar, Deborah Khider, Julien Emile-Geay, and Nicholas McKay. 2017. A Controlled Crowdsourcing Approach for Practical Ontology Extensions and Metadata Annotations. (2017), 231–246 pages.

[14] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Lam Ngoc Tran, and Karl Aberer. 2013. An evaluation of aggregation techniques in crowdsourcing. (2013), 15 pages.

[15] Panagiotis G Ipeirotis. 2010. Demographics of mechanical turk. (2010).

[16] Yuan Jin, Mark Carman, Dongwoo Kim, and Lexing Xie. 2017. Leveraging Side Information to Improve Label Quality Control in Crowd-sourcing. (2017).

[17] Hyun Joon Jung and Matthew Lease. 2011. Improving Consensus Accuracy via Z-Score and Weighted Voting.. In *Human Computation*.

[18] Ece Kamar and Eric Horvitz. 2012. Incentives for truthful reporting in crowdsourcing. In *Proceedings of the 11th international conference on autonomous agents and multiagent systems-volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, 1329–1330.

[19] Evangelos Kanoulas, Ben Carterette, Mark Hall, Paul Clough, and Mark Sanderson. 2011. Overview of the trec 2011 session track. (2011).

[20] David R Karger, Sewoong Oh, and Devavrat Shah. 2013. Efficient crowdsourcing for multi-class labeling. *ACM SIGMETRICS Performance Evaluation Review* 41, 1 (2013), 81–92.

[21] Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. 2012. The face of quality in crowdsourcing relevance labels: Demographics, personality and labeling accuracy. (2012), 2583–2586 pages.

[22] John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. 2010. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *2010 workshop on crowdsourcing for search evaluation*. 21–26.

[23] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. (2014), 1188–1196 pages.

[24] Hongwei Li, Bin Yu, and Dengyong Zhou. 2013. Error rate analysis of labeling by crowdsourcing. In *Machine Learning meets Crowdsourcing Workshop*.

[25] Hongwei Li, Bin Yu, and Dengyong Zhou. 2013. Error rate bounds in crowdsourcing models. *arXiv preprint arXiv:1307.2674* (2013).

[26] Hongwei Li, Bo Zhao, and Ariel Fuxman. 2014. The Wisdom of Minority: Discovering and Targeting the Right Group of Workers for Crowdsourcing. In *WWW*. NY, USA, 165–176.

[27] Andrew Mao, Ece Kamar, Yiling Chen, Eric Horvitz, Megan E Schwamb, Chris J Lintott, and Arfon M Smith. 2013. Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing. (2013).

[28] Afra J. Mashhadi and Licia Capra. 2011. Quality Control for Real-time Ubiquitous Crowdsourcing. In *UbiCrowd*. NY, USA, 5–8.

[29] Hayam Mousa, Sonia Benmokhtar, Omar Hasan, Lionel Brunie, Osama Younes, and Mohiy Hadhoud. 2017. A Reputation System Resilient Against Colluding and Malicious Adversaries in Mobile Participatory Sensing Applications. (2017).

[30] David Oleson, Alexander Sorokin, Greg P Laughlin, Vaughn Hester, John Le, and Lukas Biewald. 2011. Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing. *Human computation* 11, 11 (2011).

[31] Habibur Rahman, Senjuti Basu Roy, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. 2015. Task assignment optimization in collaborative crowdsourcing. (2015), 949–954 pages.

[32] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual international conference on machine learning*. ACM, 889–896.

[33] Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. (2007).

[34] Peter J Rousseeuw and L Kaufman. 1990. *Finding groups in data*. Wiley Online Library Hoboken.

[35] Senjuti Basu Roy, Ioanna Lykourentzou, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. 2015. Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal* 24, 4 (2015), 467–491.

[36] Jeffrey M. Rzeszotarski and Aniket Kittur. 2011. Instrumenting the Crowd: Using Implicit Behavioral Measures to Predict Task Performance. In *UIST*. NY, USA, 13–22.

[37] Gerard Salton and Michael McGill. 1983. Modern information retrieval. (1983).

[38] Christina Sarasua, Elena Simperl, Natasha Noy, Abraham Bernstein, and Jan Marco Leimeister. 2015. Crowdsourcing and the Semantic Web: A research manifesto. *Human Computation (HCOMP)* 2, 1 (2015), 3–17.

[39] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Conference on empirical methods in natural language processing*. Association for Computational Linguistics, 254–263.

[40] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. 2010. The multidimensional wisdom of crowds. (2010), 2424–2432.

[41] Jacob Whitehill, Ting fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. 2009. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *NIPS*. 2035–2043.

[42] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3 (2016).

[43] Bin Ye, Yan Wang, and Ling Liu. 2015. Crowd trust: A context-aware trust model for worker selection in crowdsourcing environments. (2015), 121–128 pages.

[44] Denny Zhou, Sumit Basu, Yi Mao, and John C Platt. 2012. Learning from the wisdom of crowds by minimax entropy. (2012), 2195–2203.