GRIFFITH UNIVERSITY

Answer validation for generic crowdsourcing tasks with minimal efforts

Author

Nguyen, Quoc Viet Hung, Duong, Chi Thang, Nguyen, Thanh Tam, Weidlich, Matthias, Aberer, Karl, Yin, Hongzhi, Zhou, Xiaofang

Published

2017

Journal Title

VLDB Journal

Version

Accepted Manuscript (AM)

DOI

10.1007/s00778-017-0484-3

Rights statement

© 2017 Springer-Verlag Berlin Heidelberg. This is an electronic version of an article published in The VLDB Journal, Volume 26, Issue 6, pp 855–880, 2017. The VLDB Journal is available online at: http://link.springer.com/ with the open URL of your article.

Downloaded from

http://hdl.handle.net/10072/371543

Griffith Research Online

https://research-repository.griffith.edu.au

Noname manuscript No.
(will be inserted by the editor)

Answer Validation for Generic Crowdsourcing Tasks with Minimal Efforts

Nguyen Quoc Viet Hung [1] · Duong Chi Thang [2] · Nguyen Thanh Tam [2] · Matthias Weidlich [3] · Karl Aberer [2] · Hongzhi Yin[4*] · Xiaofang Zhou [4] ·

Received: date / Accepted: date

Abstract Crowdsourcing has been established as an essential means to scale human computation in diverse Web applications, reaching from data integration to information retrieval. Yet, crowd workers have wide-ranging levels of expertise. Large worker populations are heterogeneous and comprise a significant amount of faulty workers. As a consequence, quality insurance for crowd answers is commonly seen as the Achilles heel of crowdsourcing. Although various techniques for quality control have been proposed in recent years, a post-processing phase in which crowd answers are validated is still required. Such validation, however, is typically conducted by experts, whose availability is limited and whose work incurs comparatively high costs.

This work aims at guiding an expert in the validation of crowd answers. We present a probabilistic model that helps to identify the most beneficial validation questions in terms of both, improvement of result correctness and detection of faulty workers. By seeking expert feedback on the most problematic cases, we are able to obtain a set of high quality answers, even if the expert does not validate the complete answer set. Our approach is applicable for a broad range of crowdsourcing tasks, including classification and counting. Our comprehensive evaluation using both real-world and synthetic datasets demonstrates that our techniques save up to 60% of expert efforts compared to baseline methods when striving for perfect result correctness. In absolute terms, for most cases, we achieve close to perfect correctness after expert input has been sought for only 15% of the crowdsourcing tasks.

[1] Griffith University
[2] École Polytechnique Fédérale de Lausanne
[3] Humboldt-Universität zu Berlin
[4] The University of Queensland
[*] Corresponding author

1 Introduction

Crowdsourcing has attracted much attention from both academia and industry, due to the high availability of Internet users (a.k.a. crowd workers) [53]. It has proved to be an efficient and scalable approach to overcome problems that are computationally expensive or unsolvable for machines, but rather trivial for humans. The number of crowdsourcing applications is tremendous, ranging from data acquisition [1], data integration [78], data mining [66], information extraction [17], to information retrieval [75]. To facilitate the development of crowdsourcing applications, more than 70 crowdsourcing platforms such as Amazon Mechanical Turk (AMT) and CrowdFlower have been developed in recent years.

A common crowdsourcing setup features users that post tasks in the form of questions, which are answered by crowd workers for financial rewards. Depending on the type of question, different types of crowdsourcing tasks are distinguished: In *discrete* tasks, workers are asked to assign one or more labels to each object that needs to be processed [23]. An example for such a task is sentiment annotation, where workers label movie reviews with two labels: positive or negative. In *continuous* tasks, objects are assigned real values, e.g., scores or measurement values [72]. An assessment of the relevance of the result for Web search is an example for such a continuous task. Furthermore, tasks may also define objects as rules (referred to as *partial-function* tasks) [1, 44] or require the evaluation of matches between entities (referred to as *similarity* tasks) [19]. An example of a partial-function task is discussed in [1], where crowd workers shall provide association rules between pre-defined items.

Much work on crowdsourcing focused on discrete tasks, also known as *classification* tasks as they are the core of many applications, such as training classifiers [66] and entity extraction [9]. In recent years, however, non-classification tasks gained increasing importance, e.g., when counting objects in an image or defining a bounding box around an object [72].

Validation of crowd answers. Regardless of the type of crowdsourcing tasks, quality control is a major obstacle. Workers have different backgrounds and wide-ranging levels of expertise and motivation [31], so that the collected answers are not always correct. To overcome this issue, tasks are often assigned to multiple workers to aggregate the results. In the presence of faulty workers giving random answers, however, the aggregated answer is not guaranteed to be correct.

To increase the trustworthiness of obtained crowd answers (referred to as an *answer set*), crowdsourcing platforms such as AMT include a validation phase. Crowd answers are validated against the supposedly correct answers given by a human validator (henceforth called *expert*) such as the task submitter himself. It should be noted that the notion of expert is different from domain experts or high-expertise workers [3], whose answers are still aggregated as normal workers.

Validation of answer by an expert leads to a trade-off between the verified result correctness and the invested effort. The more effort the expert puts into providing answers that can be used to judge correctness of answers from crowd workers, the higher is the quality of the final answer set. Seeking expert input incurs high costs, so that, given the sheer amount of questions to be answered, only a fraction of the answer set can be validated based on the expert's answers. In fact, validating a large part of the crowd answers would negate the benefits of crowdsourcing in the first place.

Contributions. This paper targets the effective utilization of expert efforts in the validation of crowd answers. By (I) *aggregating answers* of crowd workers and (II) *guiding an expert* in the validation process, we go beyond the aforementioned trade-off and reduce the amount of expert efforts needed to achieve the same level of result correctness. Both steps, answer aggregation and expert guidance, are interrelated. On the one hand, answer aggregation exploits the reliability of workers, which is assessed based on the feedback given by an expert as part of the answer validation. On the other hand, an expert is guided based on the potential effect that the validation of a certain answer has for the aggregation of answers.

(I) Answer aggregation. To aggregate answers obtained from crowd workers, we develop a probabilistic model estimating whether a certain answer is correct. In order to cope with diverse types of crowdsourcing tasks, our approach is based on the notion of a factor graph, which enables us to model complex relations between crowd answers, expert input, and among the labels themselves. Unlike traditional likelihood estimators which only take into account the answer set, see [23], our estimator is able to achieve higher accuracy by also considering expert input. In particular, the expert input is used to assess the reliability of a worker, captured as variables in the factor graph. The reliability of workers is then exploited to calculate the probability that a certain answer is correct. Moreover, a decision-theoretic measure allows us to conclude on the uncertainty related to an answer set based on the reliability of workers. Since expert input is sought continuously, it is important to realize answer aggregation as pay-as-you-go process. We achieve this by updating the model for worker reliability incrementally upon the arrival of new expert input.

(II) Expert guidance. To guide the validation of crowd answers by an expert, we formally define the problem of effort minimization to reach a validation goal in terms of result correctness. The problem can only be solved when assuming that workers are truthful. Even in that case, which is not realistic, however, the problem is intractable—even a restricted variant of the problem is NP-hard. Hence, we introduce two guidance strategies that cater for complementary aspects of the problem:

- The first strategy aims at a maximal improvement of the result correctness, which is motivated by the observation that some expert validations are more beneficial than others. Since workers and tasks are not independent, a certain expert input may have a positive effect on the evaluation of the worker reliability and, thus, on the estimated result correctness. We show how a measure for the expected benefit of a validation question can be used to guide an expert.
- The second strategy focuses on the detection of faulty workers (e.g. spammers), which can account for up to 40% of the worker community [31]. Faulty workers increase the cost of acquiring correct answers and contaminate the answer set by adding uncertainty. Hence, we estimate the likelihood of a worker to be a spammer and show how an expert can be guided to detect faulty workers.

Both guidance strategies have different strengths, so that we also present a hybrid approach. It combines the two strategies dynamically.

In addition, we propose various methods that allow to terminate the validation process early when the benefit of getting new feedbacks is negligible. Early termination helps to reduce the expert effort further while the quality is unchanged.

We evaluated the developed approach with multiple real-world and synthetic datasets. Our techniques save up to 60% of expert efforts compared to a baseline method when striving for perfect result correctness. For most cases, we achieve close to perfect correctness with expert input on only 15% of the questions. Also, the explicit integration of answer validations as realized by our techniques is twice as effective in increasing result correctness compared to the traditional approach of integrating expert input as crowd answers. Moreover, we demonstrate robustness of the approach against erroneous expert input and show that, by distributing a cost budget between crowd workers and experts, it achieves high correctness while satisfying completion time and budget constraints.

This paper extends and revises our previous work [43] on minimizing expert efforts in validating crowd answers. The approach presented in [43] is applicable solely for one specific type of crowdsourcing tasks, i.e., classification tasks. In these discrete tasks, labels that shall be assigned to objects are independent, so that our earlier approach is grounded in a probabilistic model based on Expectation-Maximization. In contrast, the approach presented in this work targets a broader class of crowdsourcing settings, including continuous, similarity or partial-function tasks where answer options are interrelated. We present novel methods for answer aggregation and the detection of faulty workers that take into account the dependencies between labels. In addition, we contribute a technique to measure the quality of aggregation results in order to terminate the expert validation process early.

The rest of the paper is organized as follows. Next we discuss characteristics of crowd workers and motivate the need to have their answers validated by an expert. §3 defines a formal model for crowdsourcing and gives an overview of our approach. The details on the proposed techniques are given subsequently: §4 introduces our method for probabilistic answer aggregation; §5 defines the problem of expert efforts minimization and presents heuristics to approximate a solution; §6 discusses how to terminate the validation process early and how to deal with erroneous expert input. Evaluation results are presented in §7, before we summarize related work in §8 and conclude in §9.

2 Background

A crowdsourcing example. An exemplary crowdsourcing task asks workers to count the number of objects in a picture. Table 1 illustrates a simple setup, in which five workers (W_1 - W_5) provided their answers to this task for four pictures (p_1 - p_4). The correct label assignments are shown in a separate column.

Table 1: Answers provided by 5 workers for 4 pictures

	W_1	W_2	W_3	W_4	W_5	Correct	Majority Voting
p_1	7	6	7	5	6	7	7 or 6
p_2	4	4	4	8	6	4	4
p_3	7	4	3	4	6	3	4
p_4	2	2	2	3	6	2	2

The quality of the result of a crowdsourcing task highly depends on the performance of the crowd workers. Previous studies [31] characterized different types of crowd workers to reflect their expertise: (1) Reliable workers have deep knowledge about specific domains and answer questions with very high reliability; (2) Normal workers have general knowledge to give correct answers, but make mistakes occasionally; (3) Sloppy workers have very little knowledge and thus often give wrong answers, but unintentionally; (4) Uniform spammers intentionally give the same answer for all questions; (5) Random spammers carelessly give random answers for all questions. The detail characteristics of worker types can be found in our previous work [43,54].

For the above example given in Table 1, for instance, worker W_1 would be considered a normal worker (three out of four answers are correct), W_3 is a reliable worker (all answers are correct), whereas W_5 is a uniform spammer (same answer to all questions). Especially in this counting task, W_2 requires careful treatment as two of their answers are correct, whereas the others are nearly correct (the difference between his answers and the correct ones is only 1, which should be tolerable). This illustrates that relations between labels need to be considered for an accurate evaluation of workers.

In practice, submitters of crowdsourcing tasks have limited control over the selection of crowd workers and little insights into the level of expertise and reliability of the workers that provided answers. Hence, tasks are often assigned to multiple workers. Various methods for answer aggregation and estimation of worker reliability have been proposed in the literature. However, the results of automatic methods are inherently uncertain, since they are heuristic-based and no technique performs well in the general case [22]. Although some techniques might achieve reliable results in a few domains [65,76,8,64], they often rely on pre-processing selection of workers (e.g. bootstrapping questions) or domain-specific heuristics, which are not always available in limited budget or cross-domain scenarios [39].

The example in Table 1 illustrates an inconsistent label assignment due to different levels of expertise of workers. For instance, three different answers are assigned for picture p_1, whereas four possible labels are provided for picture p_4. Yet, the popular approach of aggregating results by 'Majority Voting' would return only a partially correct result for the example.

Validation of crowd answers. To overcome the inherent uncertainty of crowdsourcing results, many crowdsourcing platforms such as AMT include a validation phase as depicted in Figure 1. This process features a validator (also referred to as a validating *expert*) that provides trustworthy answers. The integration of trustworthy input from experts is, in many cases, more efficient than enlarging the number of considered crowd workers. In fact, our evaluation in §7 shows that the inclusion of expert input, even though it is more expensive then additional input by crowd workers, is preferable in all but extreme cases (e.g., when the expert is more than 100 times more expensive than crowd workers).

Expert input is commonly considered to be correct, not only in crowdsourcing, but also in the related fields of error correction in databases [74] or active learning [2]. Expert input provides a ground truth for the assessment of the crowd answers. As part of our evaluation in §7, we empirically show that it is indeed reasonable to assume that experts provide correct answers. Yet, we later also investigate cases where expert input may include a certain amount of incorrect answers.

Although most crowdsourcing platforms acknowledge the need for validation, they only provide rudimentary support for the validation phase. The state-of-the-art in answer validation confronts the validating expert with the raw answer set data, complemented by simple statistics of the distribution of answer values [69, 5]. As such, the process of aggregating and validating answers from crowd workers is largely *unguided*. This is an issue given that the effort budget for validation is limited and, without guidance, validation effort is likely to be wasted on answers that have a limited potential for increasing the correctness of the overall result.

For the example in Table 1, the validation of 4 being the correct label for object p_2, for instance, would allow for assessing workers W_1, W_2 and W_3 as reliable. Feedback on picture p_3 would be more beneficial, though, as it helps to identify W_3 as a reliable worker, who indeed labelled all objects correctly.

Against this background, our work is the first to propose a method for post-processing answer validation that combines crowd answers with expert feedback for pay-as-you-go quality control in a generic setting. With the goal to minimize validation efforts, we get the best of both worlds: the cost of crowdsourcing is lower than having an expert answering all questions, whereas answer validation increases the result correctness.

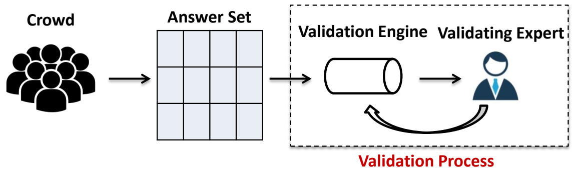

Fig. 1: A simple answer validation process

3 Model and Approach

This section presents a crowdsourcing model and, based thereon, gives an overview of our overall approach to answer validation.

3.1 Model

As detailed above, crowdsourcing tasks can be classified into four different types: discrete, continuous, partial-function and similarity tasks. We generalize the differences of these task types by introducing a notion of label similarity to capture dependencies between labels. For example, for continuous tasks, the space of labels is discretized and ordering between the discrete values is realized by defining their similarity. For partial-function tasks, such as association rule aggregation and ranking aggregation, we can consider each association rule or rank as a label, again resorting to the notion of similarity to induce an order between them. Similarity tasks, in turn, define a similarity measure explicitly, which can be lifted to possible labels.

Against this background, we formalize crowdsourcing as follows. The input of our model consist of a set of k workers $W = \{w_1, \ldots, w_k\}$ that provide answers for a set of n objects $O = \{o_1, \ldots, o_n\}$. Also, there is a set of labels $L = \{l_1, \ldots, l_m\}$ and a function $sim : L \times L \to [0, 1]$ that measures the similarity between the labels. The similarity function allows us to model different types of tasks. For instance in case of discrete tasks, the similarity values between the labels are 0. Then, crowd answers are modelled as an $n \times k$ *answer matrix*:

$$\mathcal{M} = \begin{pmatrix} x_{11} & \ldots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{nk} \end{pmatrix}$$

where $x_{ij} \in (L \cup \{\ominus\})$ for $1 \leq i \leq n$, $1 \leq j \leq k$. Here, the special label $\ominus$ denotes that a worker did not assign a label to an object. We write $\mathcal{M}(o,w)$ to denote the answer of worker w for object o.

Using the above notions, we define an *answer set* as a quadruple $N = \langle O, W, L, \mathcal{M}\rangle$ where O is a set of objects, W a set of workers, L a set of labels, and $\mathcal{M}$ an answer matrix.

Expert input is modelled by an *answer validation function* $e : O \to (L \cup \{\ominus\})$ that assigns labels to objects. Again, the label $\ominus$ denotes that the expert has not yet assigned a label to an object.

Our model includes the reliability of workers by means of a *confusion matrix* over labels. For a worker $w \in W$ and a set of labels $L = \{l_1, l_2, ..., l_m\}$, there is an $m \times m$ confusion matrix $\mathcal{F}_w$, such that $\mathcal{F}_w(l,l') \in [0,1]$ denotes the probability that the worker w assigns the label l' to an object for which the correct label is l.

Further, our work employs a probabilistic aggregation of crowd answers. For each combination of a label and an object, our model includes an assignment probability. For $O = \{o_1, o_2, ..., o_n\}$ as the set of objects and $L = \{l_1, l_2, ..., l_m\}$ as the set of labels, a probabilistic assignment is captured by an $n \times m$ *assignment matrix* $\mathcal{U}$. Here, $\mathcal{U}(o,l) \in [0,1]$ denotes the probability that $l \in L$ is the correct label for object $o \in O$ and we require that the matrix defines a probability distribution for each object, i.e., $\sum_{l \in L} \mathcal{U}(o,l) = 1$.

Combining the above notions, a *probabilistic answer set* is a quadruple $P = \langle N, e, \mathcal{U}, \mathcal{C}\rangle$ where $N = \langle O, W, L, \mathcal{M}\rangle$ is an answer set, e is an answer validation function, $\mathcal{U}$ is an assignment matrix, and $\mathcal{C} = \bigcup_{w \in W}\{\mathcal{F}_w\}$ is a set of confusion matrices.

The actual result of the crowdsourcing process is a *deterministic assignment*, a function $d : O \to L$ assigning labels to objects.

3.2 The overall approach to answer validation

Validation happens iteratively, such that in each step, an expert asserts the correct label for an object. This process halts either when reaching a *validation goal* or upon consumption of an *expert efforts budget*. The former relates to the desired quality of the result assignment, e.g., a threshold on the estimated correctness of the deterministic assignment. Since expert input is a scarce resource, the latter defines an upper bound for the number of validations and, thus, iterations of the validation process.

Starting with an answer set $N = \langle O, W, L, \mathcal{M}\rangle$, the validation process continuously updates a deterministic assignment, which is considered to be correct. Each iteration of the process comprises the following steps:

(1) *select* an object o for which expert feedback shall be sought;

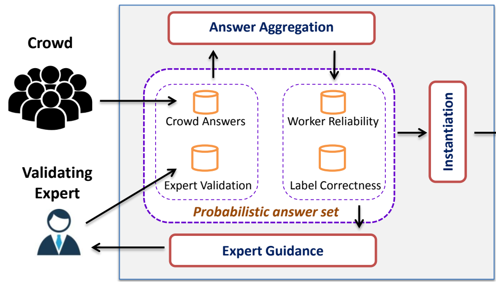

Fig. 2: Framework for guided answer validation

(2) *elicit* expert input on the label of object o and update $e(o)$,

(3) *conclude* the consequences of the expert input on the probabilistic answer set P;

(4) *filter* the deterministic assignment d assumed to be correct based on the probabilistic answer set P.

Instantiations of the general validation process differ in their implementation of steps (1), (3), and (4). For instance, a simple manual validation process is emulated as follows: an object is randomly *selected*; as part of the *conclusions*, the probability of the object for which feedback has been sought is updated; *filtering* selects, for all objects, the labels with highest assignment probability.

Striving for guided answer validation, an overview of our approach is presented in Figure 2. An initial answer set is built from the workers' responses, which is then used to construct a probabilistic answer set by means of *Answer Aggregation* under consideration of the worker reliability. Based on a probabilistic answer set and the input sought from the validating expert, we can automatically derive a deterministic assignment to be used by crowdsourcing applications, which is referred to as *Instantiation*. The quality of the deterministic assignment depends on the degree of uncertainty in the probabilistic answer set. This uncertainty stems from the decision whether to trust certain workers and select their answers when computing the assignment. *Expert Guidance* helps to resolve the uncertainty by selecting and ranking candidate objects to seek expert input. This closes the cycle since the answer validation leads to a new assessment of the worker reliability and, thus, a new probabilistic answer set. Hence, the probabilistic answer set is updated in a pay-as-you-go process, where a deterministic assignment can be instantiated at any time.

There is the following relation between the components of the framework as visualised in Figure 2 and the validation process:

Answer Aggregation. This component assesses the reliability of workers and, based thereon, computes a probabilistic assignment of labels to objects. As such, it corresponds to step *conclude* in the validation process and creates the probabilistic answer set. The realization of this component is detailed in §4.

Expert Guidance. To guide the uncertainty reduction step, this component selects and ranks objects for which expert feedback should be sought. Hence, this component realizes step *select* in the validation process, for which the details are given in §5.

Instantiation. This component creates the deterministic assignment from the probabilistic answer set, realizing step *filter* in the validation process. It is implemented as the selection of the label with the highest probability in the assignment matrix for each object.

4 Probabilistic answer aggregation

Given the answer set provided by the workers, a probabilistic answer set is constructed by assessing the worker reliability and computing the probabilistic assignment of labels to objects. We first describe the construction of a probabilistic answer set (§4.1) and then turn to a measure for the answer set uncertainty (§4.2).

4.1 Construction of a probabilistic answer set

In the construction of a probabilistic answer set, we consider the following aspects:

Expert validations. The expert input provides the supposedly correct labels for some of the objects. It helps not only to ensure correctness of the final deterministic assignment, but also allows for identifying reliable workers.

Worker Reliability. We expect label assignments done by reliable workers to be mostly correct, whereas unreliable workers provide mostly incorrect assignments. Yet, the level of reliability varies between workers and is not known apriori.

Label Similarity. Workers who choose labels that are similar to the correct one may also be reliable as the mistakes could be unintentional. On the other hand, workers who choose labels that are highly dissimilar to the correct one are highly unreliable.

Assignment correctness. For each combination of labels and objects, we have to consider the possibility that the respective assignment is correct. Clearly, the correctness of such an assignment is not known except for those that have been obtained from the expert, but we can expect reliable workers to provide mostly correct assignments.

There is a mutually reinforcing relationship between workers and objects: one worker can label multiple objects and one object can be labelled by multiple workers. Aiding this relationship, expert validations provide a means to judge both, the reliability of workers and the correctness of label assignments. In addition, the label similarity refrains us from using traditional answer aggregation methods, e.g., majority voting and expectation maximization [7], which focus on discrete labels. As a result, we approach the construction of a probabilistic answer set using the model of a factor graph [77]. It allows for concurrent estimation of worker reliability and assignment correctness, when there is a relation between the labels. Another advantage of a factor graph is that it enables self-configuration when new information becomes available, avoiding the need to manually tune model parameters. That is, with the arrival of new expert validation, the model is updated incrementally by adding variables and factors, instead of reconstructing it from scratch.

Creation of the factor graph. A factor graph is a bipartite graph $\langle V, F, E\rangle$ where V is a set of random variables, F is a set of functions (factors), and $E \subseteq \{\{v, f\} \mid v \in V, f \in F\}$ are undirected edges. A set of random variables V and a set of factors F fully characterises a factor graph. The definition of the edges relates each factor $f(v_1, \ldots, v_d) \in F$ to the random variables over which it is defined, i.e., $\{f, v_i\} \in E$ for $v_i \in V$, $1 \le i \le d$.

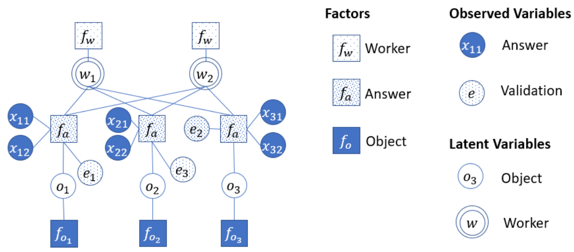

Fig. 3: An example of a factor graph

In our context, there are four types of random variables representing workers, expert validations, objects, and answers. We overload notation and use W, e, O and X to refer to the actual workers, expert validations, objects and answers, as well as to the associated random variables, i.e., $V = W \cup e \cup O \cup X$. Further, our model includes worker factors f_W, object factors f_O, and answer factors f_A to represent the relations between these variables, i.e., $F = f_W \cup f_O \cup f_A$.

Worker variables. Each worker $w \in W$ is associated with a random variable, which, overloading notation, is denoted by $w \in [0, 1]$. In fact, the worker variable encapsulates the confusion matrix $\mathcal{F}_w$ of the worker. That is, we use $\mathcal{F}_w(l, l')$ and $w(l, l')$ interchangeably to represent the probability that the worker w assigns the label l' to an object for which the correct label is l.

Object variables. Each object $o \in O$ is associated with a variable $o \in L$ indicating the actual correct label for this object. In turn, it encapsulates the assignment matrix via computing the probability $\mathcal{U}(o, l)$ that $l \in L$ is the correct label for object $o \in O$.

Answer variables. Each answer x_{ij} is also directly considered as an (observed) variable.

Expert validation variables. Expert input for a given object o is an (observed) variable $e_i \in (L \cup \{\ominus\})$ indicating the correct label for o (i.e. e is connected to o via the answer factor f_a). In the case of $\ominus$, it denotes that the object has not received an expert validation.

Worker factors. Each worker variable w is associated with a prior-distribution factor $f_w : \{w\} \rightarrow [0, 1]$ that is determined either in a training phase or stems from external sources such as the crowdsourcing service provider. If no information is available, we start with $f_w(w) = 0.5$ following the maximum entropy principle. The set of worker factors is defined as $f_W = \bigcup_{w \in W} f_w$.

Object factors. Each object variable o is also associated with a prior-distribution factor $f_o : \{w\} \rightarrow [0, 1]$ that reflects prior knowledge about the correctness of the labels of the object. If no information is available, we also follow the maximum entropy principle. The set of object factors is defined as $f_O = \bigcup_{o \in O} f_o$.

Answer factors. Each object $o \in O$ is assigned an answer factor $f_a : W \times \{o\} \times X_o \times e_o \rightarrow [0, 1]$ that captures the relation between the workers, the object, its related answers and expert validation. This factor incorporates the intuition that (1) each object has only one correct label, (2) workers who have correct answers are reliable, (3) workers whose answers are similar to the correct label are also reliable, (4) workers whose answers are highly dissimilar to the correct label are unreliable and (5) expert feedback is the most important factor if available. Against this background, the answer factor f_a is defined as:

$$f_a(w_1, \ldots, w_m, x_{1i}, \ldots, x_{mi}, o_i, e_i) = \begin{cases} \prod_{j \in \{1,\ldots,m\}} sim(x_{ji}, o_i) \times \mathcal{F}_w(x_{ji}, o_i) & e_i = \ominus \\ \prod_{j \in \{1,\ldots,m\}} sim(x_{ji}, e_i) \times \mathcal{F}_w(x_{ji}, e_i) & e_i = o_i \\ 0 & e_i \neq o_i \end{cases} \quad (1)$$

According to the above definition, the probability of a reliable worker answering correctly or nearly correctly and the probability of an unreliable worker answering incorrectly or highly incorrectly are high. On the other hand, the probability of an unreliable worker answering correctly or nearly correctly or the probability of an reliable worker answering incorrectly or highly incorrectly are low. In addition, when the expert feedback is available, the reliability of the worker and the correct labels are calculated based on the feedback. In other words, the above definition can reflect our intuition accurately.

Example 1 Figure 3 illustrates the model of a factor graph for the setting of two workers and three objects. Variables (circles) are linked to their respective factors (squares). There are two types of variables: white circles are latent variables, whereas filled circles are observed variables. For instance, factor f_{a_1} connects the observed answer variables x_{11}, x_{12}, observed expert validation e_1 and three latent variables: worker variables w_1, w_2 and object o_1. The relation between these variables are captured in Equation 1.

Probability computation. The model of a factor graph enables us to compute the probabilities of correctness of the labels and the workers. This computation exploits the (marginal) probabilities of the random variables representing the workers, the correctness of a label, and the answers. More precisely, given a worker $w \in W$, the matrix-valued random variable $w(.,.)$ assuming a certain value is given on its matrix elements as a probability distribution over $[0, 1]$ with the constraint that $\sum_{i=l, j \in L} w(i, j) = 1$; i.e. each row of the confusion matrix represents a multinomial distribution. Object variables are multivariate which takes values in L, so that for each label $l \in L$ of an object o, there is a probability $\mathcal{U}(o, l) = Pr(o = l)$ that indicates the likelihood of label l being the correct label for o. Answer variables, in turn, are observed, which means that the probability of their observed value is 1, whereas any other value has a probability of 0. Probability computation is based on the correlations defined by the factor functions that relate the random variables to each other.

To compute probabilities in a factor graph, various techniques have been proposed in the literature. *Belief propagation* considers the (un)certainty as information that is propagated through the factor graph, e.g., by message-passing algorithms or sum-product algorithms [33]. The drawback of these techniques is that they are very slow to converge if the graph is large and contains circles [77]. When applying factor graph to the crowdsourcing setting, the number of variables grows quickly, resulting in a large and dense factor graph. Therefore, we resort to *sampling* to find the most probable values of random variables, while taking into account the factors connecting them. In other words, given a joint distribution represented by a factor graph, we want to obtain k samples that approximate this joint distribution, since from these samples, statistics about the distribution can be obtained. In particular, Gibbs sampling has proved to be a highly efficient and effective mechanism for factor graphs [77]. The idea of Gibbs sampling is to sample the conditional distributions of the model, which are represented by factor nodes connecting the variables in the factor graph. Further details on Gibbs sampling can be found in [77].

4.2 The uncertainty of answer aggregation

The heterogeneity among the workers renders it likely that many objects, which are supposed to have a single correct label, are assigned to different labels by the workers. The model of a probabilistic answer set, as constructed by the factor graph model introduced above, provides us with a truthful representation of the uncertainty related to the aggregation of the answers. To guide an expert in the validation process, the uncertainty needs to be quantified.

Let $P = \langle N, e, \mathcal{U}, \mathcal{C} \rangle$ be a probabilistic answer set constructed for answer set $N = \langle O, W, L, \mathcal{M} \rangle$. Recall that P defines an assignment $\mathcal{U}(o, l)$ for each label $l \in L$ and object $o \in O$, which represents the likelihood of l to be the correct label for o. Since the probabilities of the labels form a distribution, i.e., $\sum_{l \in L} \mathcal{U}(o, l) = 1$, we can model each object o as a random variable. Then, the overall uncertainty of the probabilistic answer set is computed by the Shannon entropy [63] over a set of random variables. More precisely, the entropy of an object o is measured as follows:

$$H(o) = -\sum_{l \in L} \mathcal{U}(o, l) \times \log(\mathcal{U}(o, l)) \quad (2)$$

The entropy of an object is the basis for the computation of the uncertainty of the probabilistic answer set P. It is defined as the sum of the entropies of all objects:

$$H(P) = \sum_{o \in O} H(o) \quad (3)$$

The entropy of an object and, thus, also of the probabilistic answer set, can only be 0, if all assignment probabilities are equal to 1 or 0. If so, there is a clear separation of correct and incorrect assignments for an object or all objects, respectively.

5 Expert validation guidance

This section presents techniques to guide an expert in the validation process that reduces the uncertainty of a probabilistic answer set. We first formalize the problem of effort minimization (§5.1). As the problem can be solved only under further assumptions on crowd workers and is computationally hard, we present two heuristic solutions aiming at a maximal uncertainty reduction (§5.2) or the detection of faulty workers (§5.3), respectively. Then, we combine both heuristics (§5.4) to achieve a comprehensive strategy of expert validation guidance.

5.1 The effort minimization problem

Instantiation of the generic answer validation process described in §3.2 requires the definition of a validation goal. For the answer aggregation introduced above, a reasonable validation goal is grounded in the uncertainty measure defined in §4.2.

Given the iterative nature of the validation process, we would like to minimize the number of necessary expert interaction steps for a given goal. For an answer set $N = \langle O, W, L, \mathcal{M} \rangle$, executing the answer validation process leads to a sequence of deterministic assignments $\langle d_0, d_1, \ldots, d_n \rangle$, termed a *validation sequence*, where d_i represents the assignment obtained after the i-th iteration. Given an expert efforts budget b and a validation goal Δ, we refer to sequence $\langle d_0, d_1, \ldots, d_n \rangle$ as being *valid*, if $n \leq b$ and d_n satisfies Δ. Let $\mathcal{R}(\Delta, b)$ denote a finite set of valid validation sequences that can be created by instantiations of the validation process. Then, a validation sequence $\langle d_0, d_1, \ldots, d_n \rangle \in \mathcal{R}(\Delta, b)$ is *minimal*, if for any validation sequence $\langle d'_0, d'_1, \ldots, d'_m \rangle \in \mathcal{R}(\Delta, b)$ it holds that $n \leq m$.

Problem 1 (Expert Efforts Minimization) *Let $\langle O, W, L, \mathcal{M} \rangle$ be an answer set and $\mathcal{R}(\Delta, b)$ a set of valid validation sequences for an expert efforts budget b and a goal Δ. The problem of* expert efforts minimization *is the identification of a minimal sequence $\langle d_0, d_1, ..., d_n \rangle \in \mathcal{R}(\Delta, b)$.*

Assuming that the validation goal is defined in terms of the uncertainty of the probabilistic answer set, solving Problem 1 is challenging. First, the objects are not independent, due to the mutual reinforcing relation between workers and objects. Validating one object can affect the uncertainty of label assignment of other objects. Second, the presence of malicious workers can alter the uncertainty of the answer set, as incorrect labels can be mistreated as correct labels and vice-versa. Further, even in the absence of faulty workers, finding an optimal solution requires investigation of all permutations of all subsets (with size $\leq b$) of objects, which is intractable. Our previous work [43] outlines that even for a restricted version of the problem, finding an optimal solution is NP-hard.

5.2 Uncertainty-driven expert guidance

Our first heuristic to guide the selection of objects for validation aims at the maximal uncertainty reduction under the assumption of ethical workers. It exploits the contribution of a single validation using the notion of information gain from information theory [61].

First, we define a conditional variant of the entropy measure introduced earlier. It refers to the entropy of the probabilistic answer set $P = \langle N, e, \mathcal{U}, \mathcal{C} \rangle$, where $N = \langle O, W, L, \mathcal{M} \rangle$, conditioned on the expert input on object o. Informally, it measures the expected entropy of P under a certain expert assignment.

$$H(P \mid o) = \sum_{l \in L} \mathcal{U}(o, l) \times H(P_l) \quad (4)$$

where $P_l = \mathit{conclude}(N, e')$ is constructed by the factor graph model with $e'(o) = l$ and $e'(o') = e(o')$ for $o' \in (O \setminus \{o\})$.

To take a decision on which object to select, we assess the expected difference in uncertainty before and after the expert input for an object. The respective change in entropy is the information gain that quantifies the potential benefit of knowing the true value of an unknown variable [61], i.e., the correct label in our case:

$$IG(o) = H(P) - H(P \mid o) \quad (5)$$

The information gain allows for selection of the object that is expected to maximally reduce the uncertainty of the probabilistic answer set in one iteration of the validation process. This is formalized by a selection function for uncertainty-driven expert guidance:

$$\mathit{select}_u(O') = \arg\max_{o \in O'} \; IG(o) \quad (6)$$

5.3 Worker-driven expert guidance

Uncertainty-driven expert guidance as introduced above assumes that workers are ethical, an assumption that is often violated in practice. Recent studies found that up to 40% of the workers in a worker community may be faulty (e.g. spammers) [31]. This section thus presents a technique for expert guidance that aims at the detection of the three problematic worker types discussed in §2, i.e., uniform spammers, random spammers, and sloppy workers.

Detecting uniform and random spammers. To assess the likelihood of a worker being a uniform or random spammer, we leverage the fact that labels provided by random spammers tend to be uniformly distributed across the correct labels, whereas labels provided by uniform spammers are all the same. These tendencies are directly visible in the *confusion matrix*, whose details are briefly summarized in our previous work [43]. However, a confusion matrix neglects the relations between labels. We overcome this limitation by extending the computation of the confusion matrix with our model:

$$\mathcal{F}^*_w = \mathcal{F}_w \mathcal{D}_{sim} \quad (7)$$

where $\mathcal{D}_{sim}$ is the $|L| \times |L|$ matrix in which $\mathcal{D}_{sim}(l, l') = 1 - \mathit{sim}(l, l')$ denotes the dissimilarity between two labels. Here, the idea is that the worker answers with similar labels should be considered as similar and not separate answers.

For the spammer detection, we rely on a variant of the *spammer score* proposed in [55] to estimate the probability that a worker is a uniform or random spammer. It is based on the observation that confusion matrices that have rows with equivalent values across columns (random spammers) or a single column with values larger than 0 (uniform spammers) have similar characteristics as a rank-one matrix. Therefore, we calculate the spammer score $s(w)$ of a worker w as the distance of the confusion matrix to its closest rank-one approximation, using the Frobenius norm:

$$s(w) = \min_{\hat{\mathcal{F}}_w} \|\mathcal{F}^*_w - \hat{\mathcal{F}}_w\|_F \quad (8)$$

where $\mathcal{F}^*_w$ is the extended confusion matrix of worker w and $\hat{\mathcal{F}}_w$ is a matrix with rank one. This low-rank approximation problem can be solved via singular value decomposition [14]. We then set a threshold τ_s to filter uniform and random spammers from the population. Moreover, it is noteworthy that in [55], the confusion matrices are constructed from the labels that are estimated to be correct, which introduces a bias if this estimation is incorrect. In our case, we construct the confusion matrices only based on the answer validations.

Detecting sloppy workers. Sloppy workers tend to provide labels incorrectly, which also contaminates the answer set. One way to detect them is also based on the confusion matrix. Following the above approach for uniform and random spammers, we construct a confusion matrix using the answer validations. As the labels provided by the sloppy workers are mostly incorrect, we can calculate the error rate of the worker. The error rate of a worker (denoted as e_w) is the sum of all values not on the main diagonal of the confusion matrix weighted by the priors of the labels. If this error rate e_w is larger than a threshold τ_p, the worker is considered as a sloppy worker. It is worth noting that we can approach in a more efficient way by reusing the factor graph constructed in Section 3. In that, we perform the belief propagation over only expert-validated objects (with the same reason of avoiding bias as above). From the factor graph, we can calculate the error rate e_w of a worker by summing over the probabilities of its random variable.

Expert guidance. We exploit the detection techniques to guide the answer validation by selecting objects that will contribute to the identification of faulty workers. To this end, we measure the benefit of expert input on an object by the expected number of detected faulty workers. Formally, by $R(W \mid o = l)$, we denote the expected number of detected faulty workers, if the answer validation indicates that l is the correct label for the object o.

$$R(W \mid o = l) = \mid \{w \mid s(w) < \tau_s\} \cup \{w \mid e_w > \tau_p\} \mid \quad (9)$$

Then, the total expected number of detected faulty workers for input on o is

$$R(W \mid o) = \sum_{l \in L} \mathcal{U}(o, l) \times R(W \mid o = l) \quad (10)$$

Hence, in each iteration of the answer validation process, the worker-driven expert guidance heuristic will select the object o with the highest total expected number of detected faulty workers, formalized by the following selection function:

$$select_w(O') = \arg\max_{o \in O'} R(W \mid o) \quad (11)$$

Handling faulty workers. A naive way to handle faulty workers is to define a threshold and exclude any worker with a spammer score higher than the threshold. However, this approach may mistakenly remove truthful workers, as illustrated by the example given in Table 2. Assuming that answer validations have been obtained only for $p_1, \ldots, p_4$, the confusion matrix would indicate that worker $\mathcal{B}$ is a random spammer, even though worker $\mathcal{B}$ will answers 4 out of 6 questions correctly. Hence, workers may be excluded too early if only a few of their answers are considered in the spammer score due to a small number of answer validations.

Table 2: Answer and confusion matrix of worker $\mathcal{B}$

	p_1	p_2	p_3	p_4	p_5	p_6
Correct	a	a	b	b	a	a
$\mathcal{B}$	a	b	a	b	a	a

	a	b
a	0.5	0.5
b	0.5	0.5

We overcome this issue by only excluding the answers of suspected faulty workers from the answer set, while continuing to collect their answers. Then, as more expert input becomes available, these answers are included again once the spammer score is higher than a threshold. In other words, any of the worker answers will be eventually be included if they are truly reliable.

5.4 A combined approach to expert guidance

There is a trade-off between the application of the uncertainty-driven and the worker-driven strategies for expert guidance. Focusing solely on uncertainty reduction may lead to contamination of the truthful workers' responses by faulty workers. On the other hand, an excessively worker-driven approach is undesirable as it may increase the overall expert efforts significantly. Therefore, we propose a dynamic weighting procedure that, in each iteration of the answer validation process, helps to choose among the two strategies.

Weighting procedure. Intuitively, there are two factors which affect the choice between the strategies:

Ratio of spammers. If a high number of faulty workers is detected, the worker-driven strategy is preferred. However, as this strategy depends on expert input, it may not be effective in the beginning when the number of answer validations is small. In this case, the uncertainty-driven strategy is favored.

Error rate. The deterministic assignment d_i captures the assignments considered to be correct in the i-th iteration of the answer validation process. If d_i turns out to be mostly incorrect, we have evidence of faulty workers in the community and, thus, favor the worker-driven strategy.

We balance both factors by combining the two strategies dynamically. In the beginning, with a low number of answer validations, it is mainly the error rate of the deterministic assignment that determines which strategy to use. At later stages, the number of detected faulty workers becomes the dominant factor.

To formalize this intuition, we denote the ratio of detected faulty workers in the i-th iteration of the answer validation process by r_i. The error rate of the deterministic assignment is computed by comparing the expert input for object o in the i-th iteration with the label l that has been assigned to o in d_{i-1}, i.e., in the previous iteration. Here, we leverage the probability $\mathcal{U}_{i-1}(o, l)$ of the probabilistic answer set $P_{i-1} = \langle N, e_{i-1}, \mathcal{U}_{i-1}, \mathcal{C}_{i-1} \rangle$, $N = \langle O, W, L, \mathcal{M} \rangle$, of the $(i-1)$-th iteration of the answer validation process. Given the answer validation that assigns l to o in the i-th iteration, the error rate is computed as:

$$\epsilon_i = 1 - \mathcal{U}_{i-1}(o, l) \quad (12)$$

Algorithm 1: Hybrid answer validation process

input : an answer set $N = \langle O, W, L, \mathcal{M} \rangle$,
a validation goal Δ,
an expert efforts budget b.
output: the result assignment d.

`// Initialization`
1 $e_0 \leftarrow (o \mapsto \ominus, o \in O)$;
2 $P_0 \leftarrow conclude(N, e_0)$;
3 $d_0 \leftarrow filter(P_0)$;
4 $i, z_0 \leftarrow 0$;
5 $end \leftarrow false$;
6 **while** $not\ \Delta \ \wedge \ i \leq b \wedge \ not\ end$ **do**
  `// (1) Select an object to get feedback`
7   $x \leftarrow random(0, 1)$;
8   **if** $x < z_i$ **then**
    `// Choose the worker-driven strategy`
9     $o \leftarrow select_w(\{o' \in O \ \mid \ e_i(o') = \ominus\})$
10   **else**
    `// Choose the uncertainty-driven strategy`
11     $o \leftarrow select_u(\{o' \in O \ \mid \ e_i(o') = \ominus\})$;
  `// (2) Elicit expert input`
12   Elicit expert input $l \in L$ on o;
13   $\epsilon_i = 1 - \mathcal{U}_{i-1}(o, l)$; `// Calculate error rate` ϵ_i
  `// (3) Handle spammers`
14   Detect spammers;
15   **if** $x < z_i$ **then** Handle detected spammers ;
16   Calculate ratio of spammers r_i;
17   $z_{i+1} = 1 - e^{-\left(\epsilon_i\left(1 - \frac{i}{|O|}\right) + r_i \frac{i}{|O|}\right)}$;
  `// (4) Integrate the answer validation`
18   $e_{i+1} \leftarrow (o \mapsto l \ \wedge \ o' \mapsto e_i(o'), o' \in O, o' \neq o)$;
19   $P_{i+1} \leftarrow conclude(N, e_{i+1})$;
20   $d_{i+1} \leftarrow (o' \mapsto filter(P_{i+1}), o' \in O, e_{i+1}(o') = \ominus \wedge o' \mapsto e_{i+1}(o'), o' \in O, e_{i+1}(o') \neq \ominus)$;
21   $i \leftarrow i + 1$;
  `// (5) Compute the early termination condition`
22   $end \leftarrow early\_terminate(P_i, P_{i+1}, d_i, d_{i+1}, e_i, e_{i+1})$;
23 return d_i;

Using the ratio of detected faulty workers r_i and the error rate ϵ_i, we compute a normalized score ($\in [0, 1]$) for choosing the worker-driven strategy:

$$z_i = 1 - e^{-(\epsilon_i(1-f_i)+r_i f_i)} \tag{13}$$

where $f_i = \frac{i}{|O|} \in [0, 1]$ is the ratio of answer validations. This score mediates the trade-off between the error rate ϵ_i and the ratio of spammers r_i by the ratio of answer validations f_i. When the ratio f_i is small, the ratio of spammers has less influence and the error rate is the dominant factor. When the ratio f_i is large, the ratio of spammers becomes a more dominant factor.

Hybrid answer validation procedure. Instantiating the general answer validation process described in §3.2, the answer validation process that incorporates both uncertainty-driven and worker-driven expert guidance is defined in Algorithm 1.

Selection of an object for which expert feedback shall be sought is done either by the worker-driven or the uncertainty-driven selection strategy ($select_w$ or $select_u$). The actual choice is realized by comparing factor z_i to a random number (line 8), thereby implementing a roulette wheel selection [18]. Thus, even if factor z_i assumes a large value, there is a chance that the uncertainty driven strategy is chosen. For each selection strategy (worker-driven at line 9 and uncertainty-driven at line 11), we consider the set of objects that have not been validated by the expert. This concludes the first step of our algorithm.

As a second step (line 12-13), we elicit the expert feedback for the object selected in the first step (line 12). Based on the validation by the expert, the error rate is computed (line 13) following Equation 12.

Next, as a third step (line 14-17), we focus on the handling of spammers. First, we run the method for detecting faulty workers (line 14). The workers detected in this step are handled if the worker-driven strategy had been selected (line 15). Further, the ratio of unethical workers r_i is calculated to compute score z_{i+1} (lines 16-17), used in the next iteration to choose between the selection strategies.

The aim of the fourth step (line 18-21) is to integrate the feedback and update the probabilistic model. Feedback is integrated by updating the answer validation function e_{i+1} (line 18). For all objects for which an expert validation is available, the answer validation function e_{i+1} returns the expert validation. Then, we compute the probabilistic answer set P_{i+1} with the function *conclude* that implements probabilistic answer aggregation as defined in §4 (line 19). The general idea is to update the factor graph with the new expert validation and recompute the probabilistic model. Next, we update the deterministic assignment set function d_{i+1} (line 20). This is done as follows: for objects for which no expert input has been received, the correct assignment is estimated based on the probabilistic answer set using the function *filter*, as discussed in §3.2. The filtered assignments, together with the answer validations, define the deterministic assignment assumed to be correct at this validation step.

Finally, we also update the early termination condition by running the *early_terminate* function (line 22), which will be discussed in §6.

Implementation. A practical implementation of the hybrid answer validation process must cope with the complexity of the computation of the information gain and the expected spammer score for each object (as part of step (1)). Therefore, to achieve an efficient implementation, we consider two techniques:

Parallelization The computations of the information gain and the expected spammer score for different objects are independent and, therefore, can be executed in parallel for all objects.

Sparse matrix partitioning Due to the implied cognitive load, workers answer a limited amount of questions. Hence, the answer matrix is sparse when having a large number of objects [26]. We use sparse matrix partitioning [30] to divide a large answer matrix into smaller dense ones that fit for human interactions and can be handled more efficiently.

6 Scalability and Robustness Considerations

Having introduced our general solution to answer validation, this section turns to scalability and robustness considerations that are relevant for any instantiations of answer validation in practice.

So far, we considered a twofold termination condition for answer validation: either the validation goal is reached or the expert budget has been utilised, see Problem 1. However, we can further improve scalability of the approach by also terminating answer validation upon convergence of the results. Intuitively, expert feedback dramatically reduces the overall uncertainty of the factor graph model in the beginning of the validation process, but its effects may become negligible at a later point in time. In that case, it is reasonable to terminate the validation process, even if the validation goal has not yet been reached and there is still some effort budget remaining. The latter may be saved without lowering the result quality. In §6.1, we discuss how to implement such early termination.

Furthermore, it is commonly assumed that the answers provided by the validating expert are correct, see §2. Yet, in practice, expert input may contain mistakes, caused not by the lack of knowledge of the expert, but stemming from the interaction as part of the validation [57]. In other words, if such erroneous answer validations are detected, they can be fixed by the expert themselves. In §6.2, we elaborate on how to handle potentially erroneous expert input and eliminate them with little extra effort.

6.1 Early-Termination of the Answer Validation Process

In Algorithm 1, we considered an option for early termination based on an explicit termination predicate *end*, for which the truth value is determined by a function *early_terminate*. Below, we consider different practical realisations of this function that indicate convergence of the answer validation process.

Uncertainty reduction rate. Our first convergence indicator is grounded in the effect of expert feedback on the uncertainty reduction. At the beginning of the answer validation process, the uncertainty of the probabilistic answer set is high, since there are many conflicting labels from the workers but little feedback from the expert. New expert input is thus highly beneficial as its information can be propagated widely to resolve many conflicts in the probabilistic answer set. While more expert feedback is received, the overall uncertainty is reduced, so that expert input has limited potential to be propagated.

Formally, after each validation step in Algorithm 1, the probabilistic answer set P_i becomes P_{i+1}. The reduction rate of uncertainty can be measured by the ratio of the uncertainty difference before and after the validation:

$$\frac{H(P_i) - H(P_{i+1})}{H(P_i)} \tag{14}$$

When the process converges, the uncertainty reduction rate reaches zero. The termination predicate *end*, thus, may be set to true when the rate levels off, e.g. $\leq 10\%$.

The number of changes. While the previous indicator is based on the probabilities of correctness of the labels, this indicator only concerns the label with the highest likelihood to be correct. This is motivated by the fact that one may be interested in the deterministic assignment rather than the probability values themselves. The purpose of this metric is to measure the change of the deterministic assignment in two consecutive feedback iterations. In some cases, the overall uncertainty is reduced but the deterministic assignment remains unchanged. For example, assume that for some object o it holds $Pr(o = T) = 0.8$ and $Pr(o = F) = 0.2$. If after integrating expert input, we have $Pr(o = T) = 0.9$ and $Pr(o = F) = 0.1$, the uncertainty of object o is indeed lower, but its most probable label remains unchanged.

After a considerable number of iterations, if the number of changes is zero or insignificant, we can conclude that the deterministic assignment is likely to be correct. Formally, after each iteration of Algorithm 1, the deterministic assignment d_i becomes d_{i+1} then the number of changes in the deterministic assignment can be measured by:

$$\sum_{o \in O} \mathbb{1}_{d_i(o) \neq d_{i+1}(o)} \tag{15}$$

The termination predicate *end* may be set to true, if the number of changes is less than a pre-defined threshold (e.g. 10%) within a number of (consecutive) iterations.

The number of good predictions. Another useful indicator for a high quality answer set is the ability to instantiate label assignments that are matched with expert input. Intuitively, if the instantiated label matches the label assigned by an expert, the probabilistic model is in good state, regardless of its level of uncertainty. Formally, in each iteration of Algorithm 1, the model derived a correct label assignment if:

$$e_i(o) = d_i(o) \tag{16}$$

Again, this information can be used to set the termination predicate *end*. For instance, it is set to true if more than a pre-defined number of (consecutive) correct label assignments are derived.

Precision Improvement Rate. The above measures are 'indirect' indicators of the process convergence. A direct way to measure the convergence is to rely on the precision of the model itself. However, in order to compute the precision of a deterministic assignment, we need to known the ground truth, which is not available. Therefore, we propose a method to estimate the precision of the deterministic assignment based on k-fold cross validation [58]. Informally, we randomly partition the expert-validated objects into 'test' and 'training' sets, and measure the correctness of the labels of the objects in the test set constructed using the expert-validated objects in the training set.

Formally, at step i in Algorithm 1, given the set of validated objects $D = \{o \in O \mid e_i(o) \neq \ominus\}$, we divide it into k equal size partitions $D = D_1 \cup \ldots \cup D_k$. We repeat the following procedure k times: (i) consider the objects of the j-th partition D_j as non-validated, (ii) calculate $P_j = \mathit{conclude}(N, D \setminus D_j)$ and $d'_j = \mathit{filter}(P_j)$, (iii) compare the calculated labels for the objects in D_j based on d'_j with the correct labels already given by the expert to compute the 'partial' precision:

$$A_{D_j} = \frac{|\{o \in D_j \mid d'_j(o) = e_i(o)\}|}{|D_j|} \tag{17}$$

For an accurate estimation, we take the average of k runs as an overall estimation of the model precision at step i:

$$A_i = \frac{\sum_{j=1}^{k} A_{D_j}}{k} \tag{18}$$

Then, we can calculate the rate of precision improvement at step i as follows:

$$\frac{A_i - A_{i-1}}{A_{i-1}} \tag{19}$$

While conducting answer validation, the precision improvement rate should converge to zero. Yet, with more expert input available, this indicator becomes more costly to compute. In practice, therefore, this indicator may only be used periodically in order to decide on early termination of the answer validation process.

6.2 Erroneous answer validations

In order to achieve robustness of answer validation, we consider the possibility of erroneous expert input (i.e., the case that the expert gives incorrect feedback). In particular, we distinguish two cases: (1) the crowd is right, i.e., the aggregated answer is correct, whereas the expert validation is wrong; (2) the crowd is wrong, but the answer validation is also wrong. As illustrated later in our evaluation, case (1) is unlikely to happen since a validating expert is confronted with statistics about crowd answers, so that a decision to deviate from the aggregated answer is typically taken well-motivated. Case (2), however, is more likely to happen since an expert is more likely to confirm the aggregated answer than to deliberately deviate from it.

We cater for erroneous answer validations as in case (2) by augmenting the answer validation process with a lightweight confirmation check. This check is triggered after a fixed number of iterations of the validation process and proceeds as follows. At step i,

(I) For every object o for which expert input has been sought, a deterministic assignment $d^i_{\sim o}$ is constructed based on the answer set N and the expert validations e from which the expert feedback for o has been excluded.
(II) The label for object o in $d^i_{\sim o}$ is compared with the respective expert feedback $e(o)$. If $d^i_{\sim o}(o) \neq e(o)$, then $e(o)$ is identified as an erroneous answer validation as in case (2).

The intuition of this approach can be described as follows. At the i-th iteration, when the expert provides feedback for the object o, the deterministic assignment constructed at this step, d^i, gives an incorrect label for object o ($d^i(o)$ is incorrect), while the input by the expert is also incorrect ($e(o)$ is incorrect). However, at a later iteration, say the j-th step ($j > i$), we run the check for erroneous answer validations. The deterministic assignment constructed at this step, $d^j_{\sim o}$, is based on a larger collection of expert validations, as it also incorporates the input received between the i-th and the j-th step. Now, it may turn out that the deterministic assignment at the j-th step returns a different label compared to the i-th step, i.e., $d^j_{\sim o} \neq d^i$. If so, the label return by $d^j_{\sim o}$ is considered to be correct for object o, since $d^j_{\sim o}$ is supposedly more trustworthy. Later, our evaluation will demonstrate that this simple check is highly effective, which makes the answer validation process robust against erroneous expert input.

7 Evaluation

This section presents an empirical evaluation of the proposed approach using both real-world and synthetic datasets. We first discuss the experimental setup (§7.1), before turning to an evaluation of the following aspects of our solution:

- The runtime performance of the presented approach (§7.2).
- The benefits of integrating expert input as a first class citizen in the answer aggregation (§7.3).
- The performance of the factor graph method for answer aggregation (§7.4).
- The effectiveness of the detection of faulty workers (§7.5).
- The effectiveness of hybrid expert guidance (§7.6).
- The effectiveness of early termination methods (§7.7).
- The robustness of the approach when experts provide erroneous input (§7.8).
- The cost-effectiveness of expert-based answer validation (§7.9).

7.1 Experimental setup

Datasets. Our experiments have been conducted on five real-world datasets and synthetic datasets. The real-world data provides us with a realistic crowdsourcing setup by means of four micro-task problems that span different application domains, such as image processing (dataset *people (ppl)*) or sentiment analysis (dataset *product (prod)*). Statistics on the sizes of the real-world datasets are given in Table 3. The ground truth is provided by experts in the field before-hand and the validation process is simulated by taking the validation from expert input according to the used guidance strategy. We further employed synthetic datasets to explore parameter spaces and understand the influence of data characteristics on the performance of the algorithms. More details on both real-world and synthetic datasets are given in Appendix A.

Metrics. In addition to the uncertainty of the probabilistic answer set defined in Equation 3, we relied on the following measures:

Relative expert efforts (E_i) is the number of expert feedbacks i relative to the number of objects n in the dataset, i.e., $E = i/n$.

Precision (P_i) measures the correctness of the deterministic assignment at each validation step. Let $g : O \to L$ be the correct assignment of labels for all objects. Then, the precision of the deterministic assignment d_i at the i-th validation step is

$$P_i = \frac{|\{o \in O \mid d_i(o) = g(o)\}|}{|O|}.$$

Percentage of precision improvement (R_i) is a normalized version of precision as it measures the relative improvement. If the precision at the i-th validation step is P_i and the initial precision is P_0, then the percentage of precision improvement is

$$R_i = \frac{P_i - P_0}{1 - P_0}.$$

Table 3: Statistics for real-world datasets

Dataset	Domain	# Objects	# Workers	# Labels
ppl	Image processing	192	43	15
obj	Image processing	453	89	20
prod	Sentiment analysis	110	35	3
arg	Knowledge extraction	326	72	7
bb	Image tagging	108	39	2

Experimental environment. All experimental results have been obtained on an Intel Core i7 system (3.4GHz, 12GB RAM). In addition, it is worth noting that except the experiment on early termination condition in §7.7, others experiments are run without terminating the validation process early as we want to evaluate the validation process thoroughly.

7.2 Runtime performance

Since answer validation entails interactions with the expert, it should show a good runtime performance. In this experiment, we studied the effects of the number of objects on the runtime performance. The reported time is the response time of the system during one iteration of Algorithm 1, i.e., the time the expert has to wait for the selection of the next object after providing input.

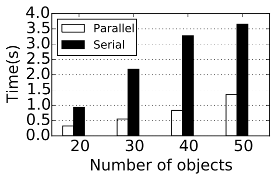

Fig. 4: Response time

#Objects	Time (s)
20	3.1
30	4.3
40	7.5
50	9.8

Table 4: Matrix ordering

Figure 4 shows the results obtained as an average of 100 runs when using matrix partitioning (see §5.4) and the plain algorithm (*Serial*) or its parallel version (*Parallel*). Increasing the number of objects from 20 to 50, which are typically found in crowdsourcing platforms [22], increases the response time. However, even for 50 objects, the response time is less than 1.5 second when using parallelization, which enables immediate interactions with humans.

Further, we evaluate the start-up time required due to matrix partitioning *before* running the actual answer validation process (which does not affect the response time for the expert). We conducted an experiment with synthetic data, 16000 questions posted randomly to 1000 workers. The sparsity of the matrix is simulated by the maximal number of questions per worker which varies from 20, 30, 40, 50. Table 4 shows that the start-up time is a few seconds.

7.3 Expert validation as first-class citizen

To study the benefits of integrating expert input as a first class citizen instead of considering it as ordinary crowd answers, we compare two ways of using expert feedback. First, each expert input is a common crowd answer in the answer aggregation (*Combined*). Second, each expert input is used to validate crowd answers as proposed in our approach (*Separate*).

Figure 5 shows the results in terms of expert effort and precision improvement for the *prod* dataset (results for other datasets are omitted as they exhibit similar characteristics). The *Separate* strategy outperforms the *Combined* strategy regardless of the expert efforts. This is expected—even though both approaches leverage the expert feedback, the precision of the *Combined* strategy is lower since expert answers are seen as equally important as those of the workers. Using the *Separate* strategy, expert input is deemed most important, overruling incorrect worker answers. As such, the results highlight the benefits of our method to integrate expert input as a first class citizen when aggregating crowd answers.

7.4 Factor graph for answer aggregation

We evaluate the factor graph model w.r.t the estimated assignment probability of the correct labels. For each object, answer aggregation should assign a higher probability to correct label than to incorrect ones. In the experiment, we keep track of the correct labels for objects and their associated probabilities while varying the expert efforts (0%, 15%, 30%). Figure 6 presents a histogram of the probability distribution in the *bb* dataset (similar results, omitted for brevity, have been obtained for the other datasets). For each object o, we measure the assignment probability $\mathcal{U}(o, l)$ of its correct label l assigned by the factor graph model. If the assignment probability of the label for object o is in a probability bin, the count for that bin is increased.

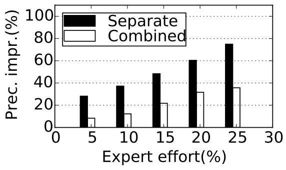

Fig. 5: Ways of integrating expert input

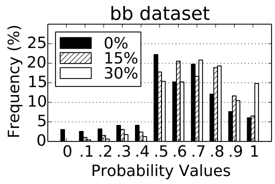

Fig. 6: Benefits of answer validation

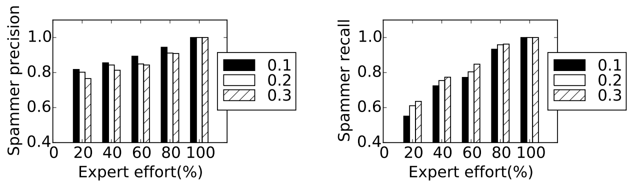

Fig. 7: Efficiency of the spammer detection technique

We note that the number of correct labels which have a probability less than 0.5 is overall small. Still, around 6% of correct labels have a probability less than 0.2 when no expert input has been integrated (expert effort 0%), meaning that answer aggregation without validation may assign a very low probability for some of the correct labels. Increasing the amount of expert input, the probability range covering most of the correct labels shifts from the 0.5 bin to higher probability bins. Hence, answer aggregation with more expert input is able to assess the assignment probabilities of the correct labels better than without expert input.

7.5 Effectiveness of the spammer detection

Since our guiding technique includes the detection of faulty workers (e.g. spammers, sloppy workers), it is necessary to analyze the technique with different detection thresholds. Since real datasets do not have information about who is spammer, we resort to using synthetic data with 20 workers that assign one of two labels to 50 objects. We then vary the threshold τ_s to detect uniform and random spammers from 0.1 to 0.3 while keeping the threshold τ_p for sloppy workers at 0.8. We also vary the validation effort from 20% to 100%. We measured the precision (ratio of correctly identified spammers over all identified spammers) and recall (ratio of correctly identified spammers over all spammers) of the detection.

Figure 7 (average of 100 runs) illustrates that, as the number of validations increases, both precision and recall of spammer detection increase. The confusion matrices used to detect spammers are built based on the answer validations. Hence, with more expert input, the confusion matrices better reflect the reliability of the workers. Also, we observe the trade-off between precision and recall as we increase the spammer score threshold. An increased threshold yields lower precision, but higher recall. Striving at a balance, we set the detection threshold to 0.2 in the remaining experiments.

7.6 Effectiveness of expert guidance

Next, we evaluate the effectiveness of our approach for reducing expert efforts on real-world datasets

Relation between uncertainty and precision. We first verified the underlying assumption of our techniques to expert guidance, i.e., that the uncertainty of a probabilistic answer set, quantified as introduced in §4.2, is correlated to the actual precision of the deterministic assignment. We perform the uncertainty-driven expert guidance on a synthetic dataset, in which we vary the number of workers from 20 to 40, the percentage of spammers from 15% to 35%, and the reliability of the workers from 0.65 to 0.75. For each combination setting of the parameters, we guide the answer validation until precision reaches 1.0 and report the uncertainty of answer aggregation along the way.

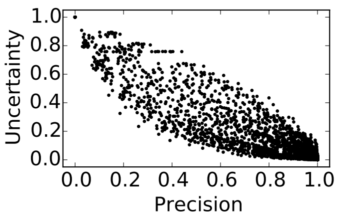

Fig. 8: Relationship – uncertainty vs. precision

Figure 8 depicts the results in terms of the relation between precision and normalized uncertainty (i.e., dividing the uncertainty values by the maximum uncertainty obtained in the run). We observe a strong correlation between both measures, which is further supported by the Pearson's correlation coefficient of -0.9257. Hence, the measured uncertainty is a truthful indicator of the result correctness.

Guidance strategies. Turning to the guidance strategies, we mimic the validating expert by using the ground-truth provided in the datasets until precision reaches 1.0. We compare the proposed approach (*hybrid*) with a method that implements the function *select* in the validation process by selecting the most 'problematic' object (*baseline*). Intuitively, we measure how 'problematic' an object is by the entropy of its probability (see Appendix B for a formal definition). This baseline method is better than random selection since it strives for the objects that are on the edge of being considered right or wrong, which are the major sources of uncertainty in the answer set.

Figure 9 shows the results for the first three real-world datasets (*ppl*, *prod*, and *arg*), the remaining dataset is discussed Appendix B. The approach developed in this paper (*hybrid*) clearly outperforms the *baseline* method. For example, in the *ppl* dataset, our approach leads to a precision above 0.9 with expert input on only 10% of the objects. The baseline method requires expert validation of around 40% to reach the same level of precision.

The relative improvement of precision for different expert effort levels is illustrated in the last plot in Figure 9. For instance, for 15% expert efforts, we achieve an improvement of precision of at least 50% for all datasets. Also, precision improvement is larger for smaller amounts of expert efforts, which emphasizes the effectiveness of our guidance strategy in particular for scenarios with a limited effort budget for the validation.

We further explored the effectiveness of our approach in relation to different aspects of a crowdsourcing setup using synthetic data. While the detailed results of these experiments are available in Appendix B, we summarize the main findings as follows. The presented approach outperforms the baseline method in terms of effectiveness (precision vs. expert effort) independent of (1) the number of possible labels, (2) the size of the crowd, (3) the worker reliability, (4) the difficulty of the questions, and (5) the presence of spammers. This indicates that the improvements obtained with the presented approach are not specific to particular crowdsourcing setups, but generalize to a wide range of applications.

7.7 Benefits of early termination

In this experiment, we study the benefits of terminating the validation process early. The experiment is conducted on the *ppl* dataset (similar results are obtained for other datasets). The following indicators are studied: (*URR*) uncertainty reduction rate – the relative difference of uncertainty of answer aggregation between two consecutive feedbacks, (*CNG*) the number of changes – the number of different deterministic assignments between two consecutive feedbacks (presented in percentage over the total number of objects), (*PRE*) the number of good predictions – the percentage of times the expert feedback agrees with the deterministic assignment (presented as histogram), and (*PIR*) precision im-

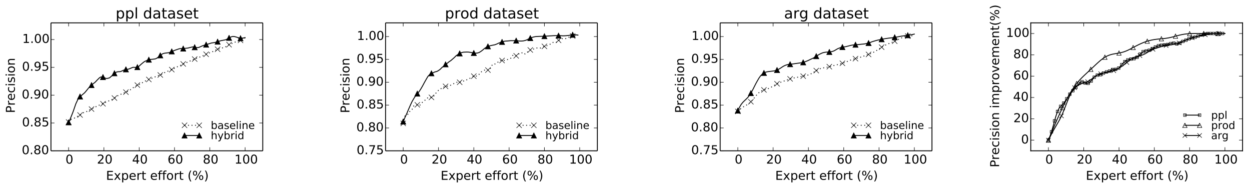

Fig. 9: Effectiveness of guiding

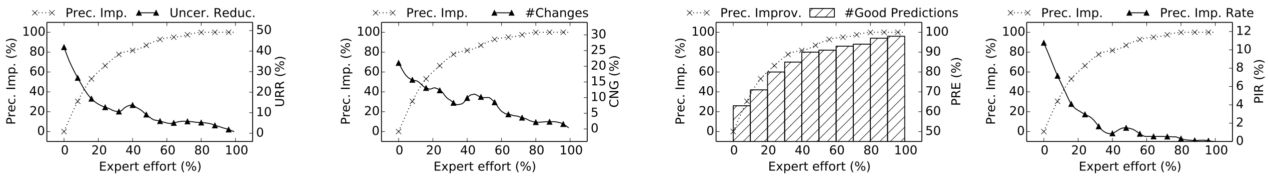

Fig. 10: Effectiveness of early termination

provement rate – the relative difference of the estimated precision between two consecutive feedbacks.

The characteristics of these indicators are shown in Figure 10. The X-axis and Y-axis are expert effort and precision improvement, respectively. The secondary Y-axis presents the values of these indicators. It can be clearly seen that the indicators are matched with the convergence status of the validation process. The expert can monitor and decide to terminate the process early for saving effort while only sacrificing small precision. For instance, using the *URR* indicator, we can stop the process early at 50% of expert effort which already achieves a large improvement of precision (over 80%). In other words, the expert can decide to terminate the process if the *URR* value is less than 10%.

7.8 Robustness against erroneous exert input

In §6.2, we discussed two types of erroneous answer validations, the expert wrongly deviates from the aggregation of crowd answers or wrongly confirms it, along with a simple confirmation check to detect mistakes of the second type.

Types of erroneous answer validations. To analyze which of the two erroneous validations is more likely to occur, we performed a preliminary study with five expert giving feedback on crowd answers for the two datasets *ppl* and *obj*. Their input was verified against the ground truth.

In general, the number of erroneous answer validations is small. For the *ppl* dataset, all experts provide correct input. For the *obj* dataset, 10% of the expert input is erroneous. For these cases, we find that, throughout, the respective answer from the crowd workers is also incorrect. This indicates that indeed, the wrong confirmation of an aggregated answer is the more likely type of mistake.

Detecting erroneous answer validations. Next, we evaluate the effectiveness of the confirmation check to detect erroneous answer validations by simulating expert mistakes. For a given probability p, we change the expert input from a correct validation to an incorrect validation. The experiment is conducted on all real-world datasets with the *hybrid* selection strategy and when triggering the confirmation check after each 1% number of total validations.

Table 5 shows the percentage of detected mistakes when increasing the probability of an expert mistake. Across all datasets, the vast majority of artificially inserted mistakes is detected. For example, even with a relatively high probability for erroneous answer validations (p=15%), all mistakes in expert input are detected.

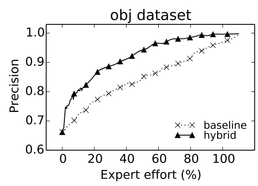

Fig. 11: Guiding with expert mistakes

Dataset	p : probability of mistake			
	0.15	0.20	0.25	0.30
ppl	100	100	92	86
obj	100	93	87	82
prod	100	100	93	88
arg	100	95	89	77
bb	100	100	94	82

Table 5: Percentage of detecting erroneous expert input

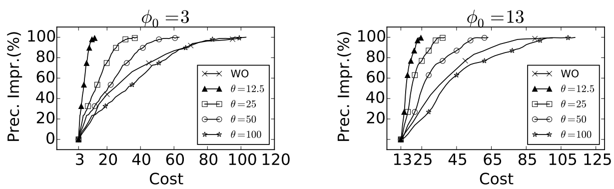

Fig. 12: Collect more crowd answers vs. validate more

Expert guidance and erroneous answer validations. Finally, we study the relation between expert effort and precision in the presence of expert mistakes. The confirmation check is run after each 1% number of total validations. Upon each detected mistake, we allow the expert to reconsider the respective input; i.e increment the expert effort by 1. The experiment is conducted using the real-world dataset *obj*, for which the experts indeed made mistakes. To aim for the worst-case scenario, we use the validations from the expert with the most mistakes.

As illustrated in Figure 11, the precision obtained with the *hybrid* strategy is still much better than the baseline method. Moreover, the actual obtained precision values are close to those obtained without erroneous answer validations (see Figure 18 in Appendix B). This result indicates that our approach is robust against potential mistakes in expert input.

7.9 Cost trade-off: Experts vs crowd workers

In the previous experiments, we have evaluated different aspects of our guiding approach for reducing expert efforts. In this final set of experiments, we aim to show that our approach is able to achieve high precision within a reasonable cost for different crowdsourcing setups. Technically, we compare two strategies: (i) *EV*, our approach that uses an expert to validate crowd answers, and (ii) *WO*, we use only the crowd and add more crowd answers with the assumption that this will increase the correctness of the answer set.

Cost model. Our cost model for this experiment covers monetary cost and completion time.

Monetary cost: We assume that the cost of an expert input is θ-times more expensive than an answer by a worker. To estimate θ, we first consider the answer cost of a crowd worker via the average wage on AMT, which is just under 2.00\$/h [59]. For the cost of an answer by an expert, we consider salary standards of traditional workers and select the most expensive case, i.e., 25\$/h, the average wage in Luxembourg [68]. Then, the ratio θ between the cost per answer of an expert and a worker is about $(25\$/h)/(2\$/h) = 12.5$.

Completion Time: Crowdsourcing in practice is often subject to a time constraint (e.g., 1 hour for simple tasks on AMT). In our setting, the completion time involves (1) *crowd time*, time for the crowd workers to answer and (2) *expert time*, time for the experts to provide input for all the questions that can be covered by the budget. Crowd time is often considered to be constant, since workers work concurrently [75]. Hence, the completion time is primarily determined by the expert time, which is derived from the number of expert inputs (assuming constant validation time for all questions).

Below, we consider a setting where m workers have been hired to answer n questions. With ϕ_0 as the average cost of asking crowd workers per object, the initial cost for deriving the answers is $n \times \phi_0$. To improve the quality of the answer set, two strategies may be followed. First, an expert can be asked to validate i answers (the EV approach), which incurs an additional cost of $\theta\times\ i$ or in total $P_{EV} = \theta \times i + n \times \phi_0$. Second, the workers can be asked to answer more questions, which increases the average cost per object to $\phi > \phi_0$. Then, the total cost of the WO approach is $P_{WO} = n \times \phi$.

Trade-off with undefined budget. In general, there is a trade-off between the cost incurred by a crowdsourcing task and the result correctness. Higher cost, spent on answer validation by an expert or additional crowd answers, yields higher correctness of the aggregated answers. We analyze this trade-off to determine under which conditions hiring only additional crowd workers (*WO* approach) is less beneficial than hiring a validating expert (*EV* approach). Figure 12 illustrates the relation between the invested cost, normalized over the number of objects ($P_{WO}/n = \phi$ and $P_{EV}/n = \phi_0 + \theta \times i/n$), and the obtained improvement in precision for different expert-crowd cost ratios $\theta = 12.5, 25, 50, 100$ and initial costs $\phi_0 = 3, 13$.

The *EV* approach yields higher precision improvements for the same costs compared to the *WO* approach with different values of ϕ_0 and for $\theta = 12.5, 25, 50$. With $\phi_0 = 3$ and $\theta = 25$, for instance, to improve the precision by 80%, the EV approach requires a cost of 22 per object, while the cost of the WO approach is 50. Also, the WO approach does not achieve 100% precision even under high costs, due to faulty workers. Having more answers from these types of workers only increases the uncertainty in the answer set.

In sum, if high precision is desired, the EV approach yields better overall results. For instance, for a realistic setup with $\phi_0 = 13$ and $\theta = 12.5$, to achieve 100% precision improvement, our approach has a cost per object of 25. The WO approach, in turn, has a cost of 100, but is still not able to achieve 100% precision improvement. When expert input is very expensive ($\theta = 100$), increasing only the number of crowd workers yields better results. However, we consider an expert-crowd cost ratio of 100 to be unlikely in practice.

Trade-off with budget constraint. The results above indicate that, without budget constraints, the *EV* approach achieves higher precision with lower cost compared to the *WO* approach regardless of ϕ_0. However, using a fixed cost, the precision obtained with the *EV* approach depends on the value of the initial cost ϕ_0. We therefore analyze how to achieve the highest precision under a fixed budget b using the EV approach. That requires deciding how much of an overall budget should be spent on retrieving crowd answers. Finding an optimal value of ϕ_0 thereby determines the best budget allocation between the expert and the crowd workers. In practice, the budget b is bounded by the cost of using only an expert, i.e., $n \leq b \leq \theta \times n$. To parameterize the budget spent on expert feedback, we formulate it as $b = \rho \times \theta \times n$, where $\rho \in [1/\theta, 1]$.

Figure 13 illustrates the result correctness in terms of precision for different allocations of the budget to crowd workers ($\phi_0/(\rho \times \theta)$), when varying the ratio ρ and setting $\theta = 25$. As a reference, the figure also includes the result for the WO approach (crowd cost is 100%), which is a special case of the EV approach where all the budget is spent on crowd workers, i.e., $\theta \times i = 0$ and $\phi_0 = b/n$.

We observe that for each ratio ρ, there is an allocation point (ϕ_0) that maximizes precision. For instance, for $\rho = 0.4$, maximal precision is obtained with 62% of the budget being spent on crowd workers and 38% of the budget used for validation by an expert. Based on this analysis, we can therefore select the optimal allocation for a specific setup. Further, except for the case with little budget ($\rho = 0.2$), a distribution of the budget between the crowd workers and the validating expert leads to maximal precision, which highlights the benefits of integrating answer validation in a crowdsourcing setup.

Trade-off with budget and time constraints. Next, we consider a setup where the best budget allocation should be determined under both, budget and time constraints. Figure 14 extends the plot of the relation between the result precision and the budget allocation with the completion time captured by the amount of expert input (y2-axis). In this figure, point B denotes the intersection between the lines representing the time constraint (green dashed line) and the completion time (orange solid line). Based on point B, a region in which the time constraint is satisfied is identified, which, in Figure 14 is bounded by the range $[C, 100]$ in terms of the allocation of the budget to crowd workers. For this region, the maximum precision is denoted by point A. As a result, we have determined the budget allocation (x-value at point A) that yields the highest precision when satisfying both the time and budget constraint.

Finally, we analyzed the effects of faulty workers, worker reliability, and question difficulty, on the cost model and the handling of the trade-off. The details of these experiments are provided in Appendix C. We found that the presented approach of using an expert to validate crowd answers, in most cases, outperforms an approach that relies solely on crowd workers. Exceptions to this trend are observed only in borderline

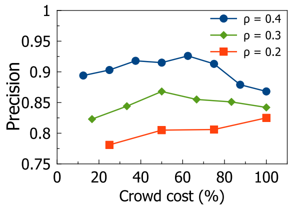

Fig. 13: Allocation of fixed budget

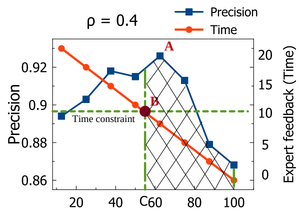

Fig. 14: Balance with time and budget constraints

cases, e.g., if the budget is extremely small (meaning that only a small number of crowd workers can be hired in the first place) and experts are much more costly than crowd workers i.e $\theta \geq 100$ (which is very unlikely in practice as this means the tasks are overpaid or too difficult even for crowdsourcing). Hence, we conclude that the integration of an expert allows for more efficient crowdsourcing for a wide range of applications.

8 Related Work

Crowdsourcing. We already discussed that crowdsourcing tasks can be categorized into four main types: discrete, continuous, partial-function, and similarity tasks. We note that all of these different types of tasks have practical relevance. Discrete tasks [23, 52, 46, 21] are used, for example, in document labelling, image tagging, and relevance feedback problems. Continuous tasks [72] are commonly found in participatory sensing and ranking problems. Partial-function tasks are the generalized form of association rule problems [1]. Finally, similarity tasks have proved valuable in record linkage and entity resolution problems [19].

One of the inherent aspects of crowdsourcing is cost, including monetary costs and completion time. Several studies focused on minimizing cost when posting tasks [4, 75, 29]. Some other works study different optimization objectives such as diversity, sparsity, and confidence [49]. In this paper, we leverage existing works on task-posting mechanism as a black-box; i.e. all of the worker answers are collected in advance before being considered by our approach. The focus of our work is the guidance for minimizing a different aspect of crowdsourcing, i.e., the cost of validating crowd answers. As a side-effect, given a limited budget constraint, our approach can predict the optimal strategy of distributing the cost for the validation and for the crowd to achieve the highest output quality (see §7.9).

Automatic quality control. Regarding quality control in crowdsourcing, there is a plethora of automatic approaches that target an assessment of the worker quality, including expertise characterization [31] and spammer detection [38]. Crowd workers can be characterized based on their level of expertise and answer strategy, for instance, as reliable workers, normal workers, uniform spammers, and random spammers [70,31]. In particular, the behaviour of spammers has received much attention and is discussed thoroughly in [10], since the proportion of spammers may be up to 40% of workers in online communities [70]. Various method to detect and control spammers have been proposed in the literature. [23] propose an EM-based algorithm to detect spammers after crowd answers have been collected. [35] use machine learning to detect spammers using Naive Bayes. In [56], the authors propose a spammer score to measure the likelihood that a worker is a spammer based on the answers given by the worker. In this paper, we propose a worker assessment mechanism that, compared to previous approaches, takes a different angle to support the expert validation process. In other words, none of the above techniques can be directly applied to our setting that incorporates expert validation [70,29,49]. Moreover, our work focuses on finding true labels via answer aggregation, rendering post-processing analysis of worker performance, such as worker profiling and disagreement analysis [65], out of scope.

Complemented with a worker assessment mechanisms, answer aggregation tries to find the hidden ground truth from the answer set given by crowd workers. Answer aggregation methods can be classified into two categories: *non-iterative* and *iterative* approaches. Non-iterative approaches [38] use heuristic methods to compute a single aggregated value of each object separately. Iterative approaches [23,29] perform a series of convergent iterations by considering the answer set as a whole. Some further techniques exploit domain-specific knowledge such as similarity between objects [6], which is not always available in our generic crowdsourcing setting. Despite the above efforts, the results of automatic quality control are inherently uncertain, since they are heuristic-based and there is no technique that performs well in the general case [22]. To address this dilemma, the semi-automatic solution presented in this paper is to employ an expert to validate crowd answers. However, none of the above aggregation methods can be directly applied to incorporate expert validation.

In addition, our answer aggregation method follows a different approach, in which we leverage a factor graph model to capture the complex relations between workers, answers, and labels. Factor graph models have been used in a crowdsourcing setting in [9,73]. Closest to our work is the work by Demartini et al. [9] that targets the entity linking problem. Despite the similarity in using the factor graph model, there are various differences between our works. First, instead of modelling the reliability of the workers as binary, we model them as continuous values, which can capture the reliability in a finer grain. Second, our factor graph also models the expert feedbacks which helps in estimating the reliability of the workers and correctness of the labels. In addition to these fundamental differences, our work is geared towards large-scale computation by relying on sampling for the probability estimation.

Crowd-Expert collaboration. While there is a large body of works on crowdsourcing, little has been done regarding incorporating expert-generated labels to the crowdsourcing tasks. Our approach is the first to *guide* an expert in the *validation* of input obtained from crowd workers. Although there are approaches for crowdsourcing that include experts, such as [20,27,28], we target a different problem setting. In particular, Karger et al. [28] rely on experts that know the reliability of crowd workers, a premise that is not realistic in the general setting for crowdsourcing explored in this work, to prove the optimality of their approach. Other works [20, 27] focus on a related, but fundamentally different problem. They target the identification of correct labels for *new* objects based on the labels for known objects, whereas we aim at validation, i.e., finding the correct labels for known objects.

Despite sharing the goal of improving the quality of crowdsourced data, our approach is orthogonal to other work on quality improvement [25,62]. While we focus on the integration of expert knowledge, it was also suggested to use pre-processing mechanisms or additional statistical information. In particular, Sarma et al. [62] decompose crowdsourcing tasks to decrease the difficulty of questions, thereby improving the chance for workers to provide correct answers. Although such decomposition is useful for large-scale data, it might render the answer matrix sparser, which requires further customization [42,48,47]. Joglekar et al. [25], on the other hand, measure the confidence interval of worker error rates, making the classification of worker types more fine-grained and thus the filtering of faulty workers more accurate.

Truth Finding. Given a set of data items claimed by multiple sources, the truth finding (a.k.a. truth discovery) problem is to determine the true values of each claimed item, with various usages in information corroboration [15], and data fusion [13]. Similar to our crowdsourcing setting, existing work on truth finding also models the mutual reinforcing relationship between sources and data items, e.g., by a Bayesian model [79], maximum likelihood estimation [71], and latent credibility analysis [51]. In contrast to our setting, these techniques incorporate prior knowledge about various aspects of the source and the data, such as the dependence between sources [11] and the temporal dimension in evolving data [12]. As such, these techniques cannot be directly applied to our solution (workers perform the tasks individually, objects do not evolve over time). To the best of our knowledge, there is no work on employing answer validation by experts to check the results of automatic techniques. Therefore, our work on guiding validation effort can be tailored to the truth finding settings as well.

Recommendation systems. Close to our work is research on recommendation systems. Here, the core problem is, given an existing set of user ratings for particular items, to recommend one of these items that best fit a particular user in terms of information content [16]. This problem is similar to ours in the sense that we also select the objects with best information content (i.e., that yield the maximal uncertainty reduction) for answer validation. However, the underlying models of the two settings are completely different. In recommendation systems, the information of an item is measured by the notion of similarity: similar users would have similar preferences on similar items and vice-versa [60]. Whereas, this similarity assumption does not exist for workers and objects in crowdsourcing. Moreover, there is a also large body of work on recommendation systems studying malicious users [36], who provide untruthful ratings or reviews to manipulate the recommendation output. Although many detection techniques have been proposed, they cannot be applied in our context since they depend on the application domains and contextual features [50]. Most importantly, there is no method making use of validation input for identifying malicious users. As a result, our work on using a validating expert to handle spammers in crowdsourcing can be tailored for recommendation systems.

Guiding user feedback. Guiding user or expert feedback has been studied in different contexts. In the field of data integration, [24] proposed a decision theoretic framework to rank candidate matches for answer validation in order to improve the quality of a dataspace. Focusing on matching of data schemas in a network setting, [45] presented a reconciliation algorithm that leverages expert input. [74], in turn, proposed an active-learning based process that requests expert input to help training classifiers in order to detect and repair erroneous data. Similar to these works, we rely on models from the fields of Decision Theory and Active Learning [61,41]. Despite the similarities in the applied models, there are two main differences between the aforementioned approaches to user guidance and the method presented here. First, in the above domains (data integration, schema matching), input stems from automatic tools, which renders it deterministic and traceable. In contrast, our methods have to cope with human input, which is unreliable, potentially non-deterministic or even malicious. Second, existing guidance methods aim at a different goal, which means that measures for the benefit of expert input are highly domain dependent (e.g., the approach in [24] is purely driven by the size of query results and independent of the source of user input). Our method is tailored to the specific characteristics of crowdsourcing scenarios.

An important problem in guiding user feedback is when to stop asking for feedback. Various works in the field of Active Learning[37,40] have studied this problem. [40] proposed a stopping condition based on an estimation of performance using held-out labels and showed that this method was able to provide reliable estimates of the quality. [37] compared various stopping conditions based on performance estimation and illustrated that there are various factors that can affect the quality of the estimation. In our work, by leveraging the factor graph, we are able to propose different stopping conditions such as uncertainty reduction rate, number of changes, which are able to show different aspects of the quality of the results.

9 Conclusions

This paper proposed techniques to support an expert in validating answers obtained for a crowdsourcing task. Based on the requirements identified for such techniques, we presented an answer validation process that features two steps: *answer aggregation* and *expert guidance*. The former relates to the creation of a probabilistic model that assesses the reliability of workers and the correctness of the crowd answers. By capturing the complex relations between the crowd, expert feedbacks and answers using a factor graph model, we are able to handle different types of crowdsourcing tasks, such as discrete, continuous, partial-function and similarity task.

We proposed different strategies for guiding an expert in the validation: worker-driven, uncertainty-driven, and a hybrid approach. The worker-driven method aims at detecting and removing faulty workers from the community, whereas the uncertainty-driven strives for a maximal improvement of the answer correctness under the assumption of truthful workers. Since both goals help to improve the overall result, our hybrid approach combines both methods with a dynamic weighting scheme. Our evaluation showed that our techniques outperform respective baselines methods significantly and save up to 60% of expert efforts. Also, in most cases, close to perfect result correctness is reached with expert input for only 15% of the considered objects.

References

1. Amsterdamer, Y., Grossman, Y., Milo, T., Senellart, P.: Crowd mining. In: SIGMOD, pp. 241–252 (2013)
2. Arasu, A., Götz, M., Kaushik, R.: On active learning of record matching packages. In: SIGMOD, pp. 783–794 (2010)
3. Callison-Burch, C.: Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In: EMNLP, pp. 286–295 (2009)
4. Cao, C.C., She, J., Tong, Y., Chen, L.: Whom to ask?: jury selection for decision making tasks on micro-blog services. In: VLDB, pp. 1495–1506 (2012)
5. CrowdFlower: http://www.crowdflower.com/ (2016)
6. Davtyan, M., Eickhoff, C., Hofmann, T.: Exploiting document content for efficient aggregation of crowdsourcing votes. In: CIKM, pp. 783–790 (2015)
7. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. J. R. Stat. Soc. pp. 20–28 (1979)
8. Dekel, O., Shamir, O.: Vox populi: Collecting high-quality labels from a crowd. In: COLT (2009)
9. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: WWW, pp. 469–478 (2012)
10. Difallah, D.E., Demartini, G., Cudré-Mauroux, P.: Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In: CrowdSearch, pp. 26–30 (2012)
11. Dong, X.L., Berti-Equille, L., Hu, Y., Srivastava, D.: Solomon: Seeking the truth via copying detection. In: VLDB, pp. 1617–1620 (2010)
12. Dong, X.L., Berti-Equille, L., Srivastava, D.: Truth discovery and copying detection in a dynamic world. In: VLDB, pp. 562–573 (2009)
13. Dong, X.L., Naumann, F.: Data fusion: resolving data conflicts for integration. In: VLDB, pp. 1654–1655 (2009)
14. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika pp. 211–218 (1936)
15. Galland, A., Abiteboul, S., Marian, A., Senellart, P.: Corroborating information from disagreeing views. In: WSDM, pp. 131–140 (2010)
16. Garcin, F., Faltings, B., Jurca, R., Joswig, N.: Rating aggregation in collaborative filtering systems. In: Proceedings of the third ACM conference on Recommender systems, pp. 349–352 (2009)
17. Gokhale, C., Das, S., Doan, A., Naughton, J.F., Rampalli, N., Shavlik, J., Zhu, X.: Corleone: Hands-off crowdsourcing for entity matching. In: SIGMOD, pp. 601–612 (2014)
18. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman (1989)
19. Gomes, R.G., Welinder, P., Krause, A., Perona, P.: Crowdclustering. In: NIPS, pp. 558–566 (2011)
20. Hu, Q., He, Q., Huang, H., Chiew, K., Liu, Z.: Learning from crowds under experts supervision. In: PAKDD, pp. 200–211 (2014)
21. Hung, N.Q.V., Tam, N.T., Miklós, Z., Aberer, K.: On leveraging crowdsourcing techniques for schema matching networks. In: DASFAA, pp. 139–154 (2013)
22. Hung, N.Q.V., Tam, N.T., Tran, L.N., Aberer, K.: An evaluation of aggregation techniques in crowdsourcing. In: WISE, pp. 1–15 (2013)
23. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on amazon mechanical turk. HCOMP, pp. 64–67 (2010)
24. Jeffery, S.R., Franklin, M.J., Halevy, A.Y.: Pay-as-you-go user feedback for dataspace systems. In: SIGMOD, pp. 847–860 (2008)
25. Joglekar, M., Garcia-Molina, H., Parameswaran, A.: Comprehensive and reliable crowd assessment algorithms. In: ICDE, pp. 195–206 (2015)
26. Jung, H.J., Lease, M.: Improving quality of crowdsourced labels via probabilistic matrix factorization. In: HCOMP, pp. 101–106 (2012)
27. Kajino, H., Tsuboi, Y., Sato, I., Kashima, H.: Learning from crowds and experts. In: HCOMP, pp. 107–113 (2012)
28. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowdsourcing systems. In: NIPS, pp. 1953–1961 (2011)
29. Karger, D.R., Oh, S., Shah, D.: Budget-optimal task allocation for reliable crowdsourcing systems. Operations Research pp. 1–24 (2014)
30. Karypis, G., Kumar, V.: Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0 (1995)
31. Kazai, G., Kamps, J., Milic-Frayling, N.: Worker types and personality traits in crowdsourcing relevance labels. In: CIKM, pp. 1941–1944 (2011)
32. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with mechanical turk. In: CHI, pp. 453–456 (2008)
33. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. TIT pp. 498–519 (1998)
34. Kulkarni, A., Can, M., Hartmann: Collaboratively crowdsourcing workflows with turkomatic. In: CSCW, pp. 1003–1012 (2012)
35. Kumar, A., Lease, M.: Modeling annotator accuracies for supervised learning. In: CSDM, pp. 19–22 (2011)
36. Lam, S.K., Riedl, J.: Shilling recommender systems for fun and profit. In: WWW, pp. 393–402 (2004)
37. Laws, F., Schätze, H.: Stopping criteria for active learning of named entity recognition. In: ICCL, pp. 465–472 (2008)
38. Lee, K., Caverlee, J., Webb, S.: The social honeypot project: protecting online communities from spammers. In: WWW, pp. 1139–1140 (2010)
39. Marcus, A., Parameswaran, A., et al.: Crowdsourced data management industry and academic perspectives. Foundations and Trends in Databases pp. 1–161 (2015)
40. Mozafari, B., Sarkar, P., Franklin, M., Jordan, M., Madden, S.: Scaling up crowd-sourcing to very large datasets: A case for active learning. In: VLDB, pp. 125–136 (2014)
41. Nguyen, Q.V.H., Do, S.T., Nguyen, T.T., Aberer, K.: Tag-based paper retrieval: minimizing user effort with diversity awareness. In: DASFAA, pp. 510–528 (2015)
42. Nguyen, Q.V.H., Duong, C.T., Nguyen, T.T., Weidlich, M., Aberer, K., Yin, H., Zhou, X.: Argument discovery via crowdsourcing. VLDBJ (2017)
43. Nguyen, Q.V.H., Duong, C.T., Weidlich, M., Aberer, K.: Minimizing efforts in validating crowd answers. In: SIGMOD (2015)
44. Nguyen, Q.V.H., Huynh, H.V., Nguyen, T.T., Weidlich, M., Yin, H., Zhou, X.: Computing Crowd Consensus with Partial Agreement. TKDE pp. 1–14 (2017)

45. Nguyen, Q.V.H., Nguyen, T.T., Miklós, Z., Aberer, K., Gal, A., Weidlich, M.: Pay-as-you-go reconciliation in schema matching networks. In: ICDE, pp. 220–231 (2014)
46. Nguyen, Q.V.H., Nguyen Thanh, T., Lam, N.T., Do, S.T., Aberer, K.: A benchmark for aggregation techniques in crowdsourcing. In: SIGIR, pp. 1079–1080 (2013)
47. Nguyen, T.T., Duong, C.T., Weidlich, M., Yin, H., Nguyen, Q.V.H.: Retaining data from streams of social platforms with minimal regret. In: IJCAI (2017)
48. Nguyen, T.T., Nguyen, Q.V.H., Weidlich, M., Aberer, K.: Result selection and summarization for web table search. In: ICDE, pp. 231–242 (2015)
49. Nushi, B., Singla, A., Gruenheid, A., Zamanian, E., Krause, A., Kossmann, D.: Crowd access path optimization: Diversity matters. In: AAAI (2015)
50. O'Mahony, M., Hurley, N., Kushmerick, N., Silvestre, G.: Collaborative recommendation: A robustness analysis. TOIT pp. 344–377 (2004)
51. Pasternack, J., Roth, D.: Latent credibility analysis. In: WWW, pp. 1009–1020 (2013)
52. Prelec, D., Seung, H.S., McCoy, J.: A solution to the single-question crowd wisdom problem. Nature pp. 532–535 (2017)
53. Quinn, A.J., Bederson, B.B.: Human computation: a survey and taxonomy of a growing field. In: CHI, pp. 1403–1412 (2011)
54. Quoc Viet Hung, N., Chi Thang, D., Weidlich, M., Aberer, K.: Erica: Expert guidance in validating crowd answers. In: SIGIR, pp. 1037–1038 (2015)
55. Raykar, V.C., Yu, S.: Ranking annotators for crowdsourced labeling tasks. In: NIPS, pp. 1809–1817 (2011)
56. Raykar, V.C., Yu, S.: Eliminating spammers and ranking annotators for crowdsourced labeling tasks. JMLR pp. 491–518 (2012)
57. Reason, J.: Human error. Cambridge university press (1990)
58. Refaeilzadeh, P., Tang, L., Liu, H.: Cross-validation. In: Encyclopedia of database systems, pp. 532–538. Springer (2009)
59. Ross, J., Irani, L., Silberman, M., Zaldivar, A., Tomlinson, B.: Who are the crowdworkers?: shifting demographics in mechanical turk. In: CHI, pp. 2863–2872 (2010)
60. Rubens, N., Kaplan, D., Sugiyama, M.: Active learning in recommender systems. In: Recommender Systems Handbook, pp. 735–767. Springer (2011)
61. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education (2003)
62. Sarma, A.D., Jain, A., Nandi, A., Parameswaran, A., Widom, J.: Surpassing humans and computers with jellybean: Crowd-vision-hybrid counting algorithms. In: HCOMP (2015)
63. Shannon, C.E.: A mathematical theory of communication. SIGMOBILE pp. 3–55 (2001)
64. Sheng, V.S., Provost, F.: Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers. SIGKDD pp. 614–622 (2008)
65. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: EMNLP, pp. 254–263 (2008)
66. Sun, C., Rampalli, N., Yang, F., Doan, A.: Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. In: VLDB, pp. 1529–1540 (2014)
67. Surowiecki, J.: The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies. ESN (2004)
68. TRAVAIL: Global wage report 2012-13. International Labour Organization (ILO) (2012)
69. Turk, A.M.: http://www.mturk.com/ (2016)
70. Vuurens, J., de Vries, A., Eickhoff, C.: How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In: CIR, pp. 48–55 (2011)
71. Wang, D., Kaplan, L., Le, H., Abdelzaher, T.: On truth discovery in social sensing: A maximum likelihood estimation approach. In: IPSN, pp. 233–244 (2012)
72. Welinder, P., Perona, P.: Online crowdsourcing: rating annotators and obtaining cost-effective labels. In: CVPRW, pp. 25–32 (2010)
73. Wick, M., McCallum, A., Miklau, G.: Scalable probabilistic databases with factor graphs and mcmc. In: VLDB, pp. 794–804 (2010)
74. Yakout, M., Elmagarmid, A.K., Neville, J., Ouzzani, M., Ilyas, I.F.: Guided data repair. In: VLDB, pp. 279–289 (2011)
75. Yan, T., Kumar, V., Ganesan, D.: Crowdsearch: Exploiting crowds for accurate real-time image search on mobile phones. In: MobiSys, pp. 77–90 (2010)
76. Zaidan, O.F., Callison-Burch, C.: Crowdsourcing translation: Professional quality from non-professionals. In: ACL, pp. 1220–1229 (2011)
77. Zhang, C., Ré, C.: Towards high-throughput gibbs sampling at scale: A study across storage managers. In: SIGMOD, pp. 397–408 (2013)
78. Zhang, C.J., Chen, L., Jagadish, H.V., Cao, C.C.: Reducing uncertainty of schema matching via crowdsourcing. In: VLDB, pp. 757–768 (2013)
79. Zhao, B., Rubinstein, B.I., Gemmell, J., Han, J.: A bayesian approach to discovering truth from conflicting sources for data integration. In: VLDB, pp. 550–561 (2012)

A Further details on Datasets

Real-world data and task design. We have used real-world datasets from different domains, namely *people (ppl)*,*object (obj)*, *product reviews (prod)*, *argument (arg)*, and *bluebird (bb)*. Opting for a generic crowdsourcing setting, our task design uses the default multiple-choice question template from AMT [32,23]. Further complex, yet out of scope, task designs aiming for human factors and exploiting domain-specific knowledge can be found in [34]. In the *ppl* dataset, workers have to count the number of people in an real-life image. The crowdsourcing tasks of the *obj* dataset comprise counting the number of people in digital-art picture. However, the questions of the *obj* dataset are more difficult than the questions of the *ppl* dataset as the people in digital-art picture are harder to recognize. In the *prod* dataset, workers are asked to annotate whether a review expresses positive, neutral or negative meaning. The tasks for the *arg* dataset require the crowd workers to extract claim and evidence related to a topic from articles from the web. In the *bb* dataset, workers have to identify one of two types of birds in an image. The similarity function – input of our model – are simply computed by uniformly normalizing the labels into natural number space. The ground truth / expert validation is provided by experts in the field.

Synthetic data. We used several generated datasets. Since this data should exhibit similar characteristics as real-world data, we considered several parameters for the data generation, in particular: (i) n – the number of objects, (ii) k – the number of workers, (iii) m – the number of labels, (iv) r – the reliability of normal workers, reflecting the probability of their answers being correct, (v) σ – the percentage of spammers in the worker population and (vi) sim – the similarity between labels, simulated as a uniform distribution in $[0, 1]$. For the synthetic dataset, we also simulated the ground truth (the correct labels) for the objects. However, it is not known by the simulated workers and only used to simulate the answer validations.

An important part of our synthetic data is the crowd simulation. We follow a practical guideline [22] to simulate the different worker characteristics of the crowd. Specially, we distribute the worker population into $\alpha\%$ reliable workers, $\beta\%$ sloppy workers and $\gamma\%$ spammers. According to a study on crowd population at real-world crowdsourcing services [31], we assign the default values of these parameters as follows: $\alpha = 43$, $\beta = 32$ and $\gamma = 25$. In the experiments, the distribution of the worker types is the same as discussed unless stated otherwise.

B Evaluations of Expert Guidance (cont'd)

In the following experiments, we analyze the effects of the guiding strategy with different crowdsourcing setup, including the number of labels, the number of workers, worker reliability, question difficulty, and the presence of spammers. Since these experiments (except the experiment on question difficulty) require changing the workers' characteristics (which is not known for the real-world datasets), they are conducted using synthetic data.

We compare the results obtained with our guiding approach (*hybrid*) to a baseline guiding method that selects the object with the highest uncertainty to seek feedback (*baseline*):

$$select(O) = \arg\max_{o \in O} H(o)$$

Our *hybrid* approach is different from the baseline as it further considers the consequences of validation in addition to the mutually reinforcing relations between the reliability of workers and assignment correctness.

Effect of the number of workers. The idea behind crowdsourcing is that individual crowd answers complement each other. Thus, the aggregation of answers should be closer to the truth as more workers participate [67]. To evaluate the effect of the number of workers on the performance of our approach, we rely on a synthetic dataset containing 50 objects. We vary the number of workers k from 20 to 40 that assign one of three labels to the objects. Figure 15 illustrates an important finding that our approach leads to better results for any number of workers. Taking a fixed amount of expert input, precision increases if more workers are employed. The reason is the widely quoted 'wisdom of the crowd' [67], which eventually leads to better precision. Another finding is that the precision improvement with the same amount of expert input is higher if we have more workers (most right plot in Figure 15). This is expected since, by having more workers, we acquire more answers for the same question, which results in better estimates of assignment probabilities and worker reliabilities. Our approach, thus, has a higher chance to select the objects that lead to a large gain in correctness.

In sum, the two findings suggest that increasing the number of workers is beneficial not only for computing assignment probabilities, but also for guiding answer validation. For the remaining experiments, we fix the number of workers to be the smallest tested value ($k = 20$), which is the most challenging scenario.

Effect of worker reliability. We further explored the effects of the worker reliability r on the effectiveness of our approach. As above, we used a dataset of 20 workers assigning one out of three labels to 50 objects. We then varied the reliability of the non-spammer workers from 0.65 to 0.75.

Figure 16 illustrates a significant improvement in precision using our approach (*hybrid*) compared to the *baseline* method. For instance, if the average worker reliability is 0.7, to achieve a precision of 0.95, our approach requires expert input for 20% of the objects, whereas the baseline method requires input for 50% of the objects. In other words, the amount of efforts the baseline method requires is 2.5 times that of our approach. Also, with the same amount of feedback, precision is increased if the average reliability of the workers is higher (most right plot in Figure 16). This is because an answer set provided by reliable workers requires less validation than an answer set coming from unreliable workers.

Effect of spammers. In this experiment, we studied the robustness of our guiding approach to spammers using the same dataset as in the previous experiment (20

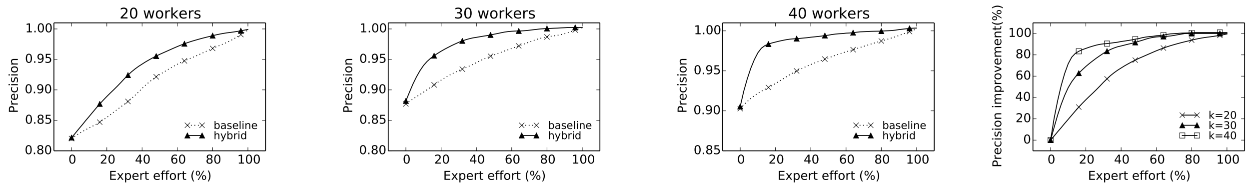

Fig. 15: Effect of number of workers

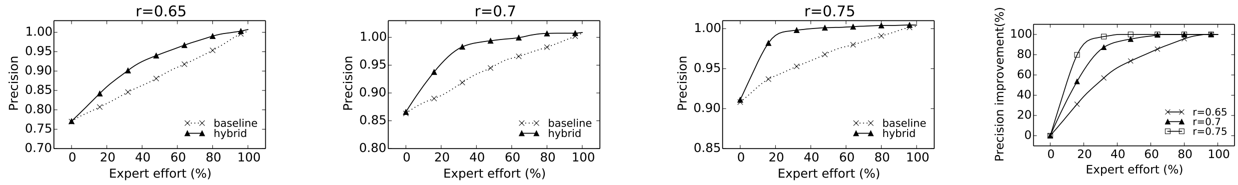

Fig. 16: Effect of worker reliability

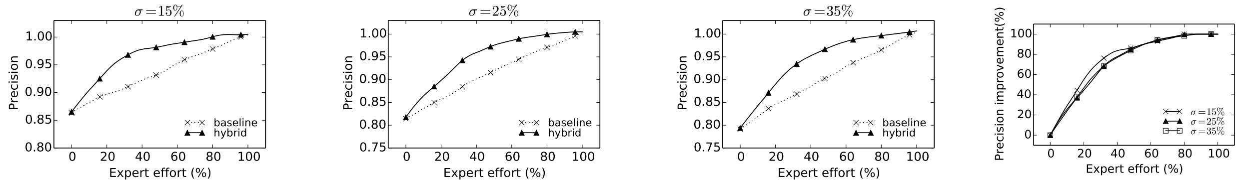

Fig. 17: Effect of spammers

workers, three labels, 50 objects). We varied the percentage of spammers σ in the worker population from 15% to 35% to analyze the effect of these spammers.

Independent of the percentage of spammers, our approach (*hybrid*) outperforms the *baseline* method, see Figure 17. The largest difference between the two approaches is observed when the percentage of spammers is 15%. In that case, to achieve a precision of 0.95, our approach needs 20% of expert input, while the baseline method requires 50%. Regarding the precision improvement (right most plot in Figure 17), the results are relatively similar across different percentages of spammers. For instance, using 50% of expert input, we are able to increase the precision of the deterministic assignment by 80%, independent of the percentage of spammers. Hence, our approach is indeed robust to the presence of spammers.

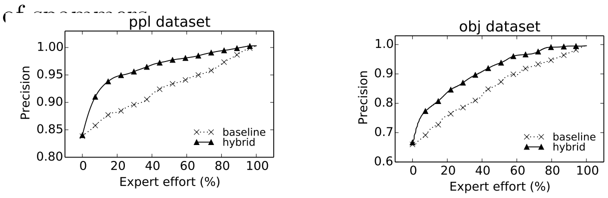

Fig. 18: Effects of question difficulty

Effects of question difficulty. Beside worker reliability, another factor that can affect the performance of our method is the question difficulty. For hard questions, even reliable workers may give incorrect answers. As a result, there is a need to analyze the effects of question difficulty on the performance of our approach.

We compared our approach with the baseline approach using two datasets: *ppl* and *obj*, where the questions in the *obj* dataset is harder than the other. The experimental results are shown in Figure 18, where the x-axis depicts the expert efforts while the y-axis illustrates the precision of the deterministic assignment.

We observe that our approach is able to outperform the baseline approach for both datasets, meaning that the approach is robust against question difficulty. For instance, for the *ppl* dataset with easy question, our approach needs only 20% of expert effort to achieve a precision of 0.95 while the baseline approach needs over 60% of expert efforts. Also, the performance of our approach when the questions are easy is better than in the setup with hard questions. This is expected and can be explained as follows. In the dataset with easy questions, most of the workers are able to give the correct answers, which makes the uncertainty in the dataset low. As a result, with the same amount of feedbacks, we can improve the precision higher than when the questions are hard.

C Cost trade-offs (cont'd)

We complement the experiments reported in §7.9 by studying the effects of question difficulty, spammers, and worker reliability when comparing the *EV* approach with the *WO* approach.

Effects of question difficulty. In this experiment, we compare our EV approach with the WO approach with respect to the difficulty of the questions. We remove the answers from the answer matrix randomly such that 13 answers remain per question ($\phi_0 = 13$). Then, to simulate the addition of answers for the WO approach, we add the answers back to the questions. We fix the expert-crowd cost ratio to $\theta = 25$ and average the results over 100 experiment runs.

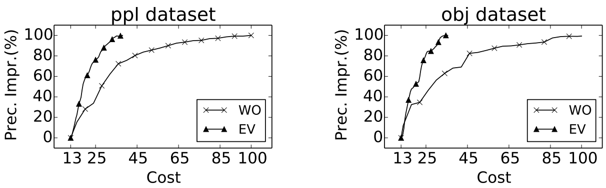

Fig. 19: Effect of question difficulty on cost

The experimental results are shown in Figure 19 where the X-axis depicts the normalized cost and the Y-axis measures the precision improvement of the deterministic assignment. The precision improvement of the EV approach is always higher than that of the WO approach, indicating that our EV approach is robust against the effects of question difficulty.

Effects of spammers. In this experiment, we analyze the effects of spammers by varying the percentage of spammers in the dataset from 15% to 35%. The experiment is conducted on the synthetic dataset with $\phi_0 = 13$, $\theta = 25$.

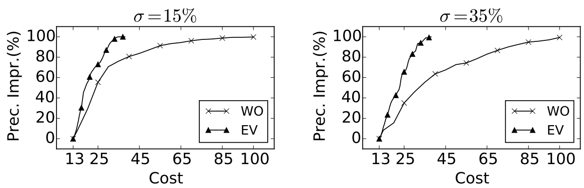

Fig. 20: Effects of spammers on cost

The results illustrated in Figure 20 show the benefits of using our approach with different percentages of spammers. The EV approach is able to achieve high precision improvement with a small amount of cost. For instance, when $\sigma = 35\%$, to improve the precision by 80%, a cost of 30 is required for the EV approach while the WO approach needs twice the amount. Also, the more spammers are part of the population, the better becomes the performance of the EV approach regarding the WO approach. For example, the difference in cost to achieve 80% precision improvement is about 15 when the percentage of spammers is 15%, but this increases three times to 30 as the percentage of spammers increases to 35%. Again, the reason is that as the percentage of spammer increases, the WO suffers from adding more answers as they are more likely to come from unreliable workers.

Effects of worker reliability. Worker reliability can affect the quality of crowd answers, thus also affects the cost model. If the worker reliability is high, the expert can spend less effort to give feedbacks as most of the answers are already correct. On the other hand, when the worker reliability is low, more feedbacks from the expert is required to achieve the same amount of precision. In this experiment, we analyze the effects of worker reliability on the cost of validating the crowd answers by varying the reliability of the normal workers from 0.6 to 0.7. Similar to the above experiment, we fix the following parameter: $\phi_0 = 13$, $\theta = 25$ and the workers population is simulated as discussed in §7.1.

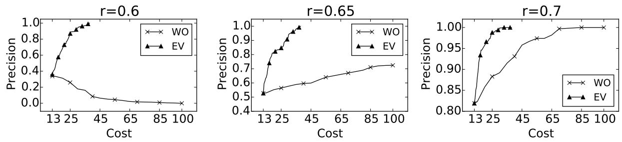

Fig. 21: Effects of worker reliability on cost

The obtained results are illustrated in Figure 21, which highlights the relation between the cost normalized over each question and the precision of the deterministic assignment. Interestingly, when the reliability of the workers is 0.6, the precision of the deterministic assignment using the WO approach converges to 0 as we add more answers. The reason is that as we decrease the worker reliability, the average worker reliability becomes less than 0.5, which makes the precision converge to 0. This shows that adding more answers to the answer set may not improve but reduce the quality due to unreliable workers. When the reliability of the workers is 0.65, the precision of the deterministic assignment using the WO approach improves very slowly as the average reliability of the whole population is about 0.5. On the other hand, when the reliability of the workers is 0.7, the precision of the WO approach converges to 1. Yet, it requires higher cost to reach the same amount of precision as the EV approach. In summary, this experiment shows that our approach is robust against the reliability of the workers.