Supporting Requesters in Writing Clear Crowdsourcing Task Descriptions Through Computational Flaw Assessment

ZAHRA NOURI, Department of Computer Science, Paderborn University, Germany
NIKHIL PRAKASH, Khoury College of Computer Sciences, Northeastern University, USA
UJWAL GADIRAJU, Web Information Systems, Delft University of Technology, Netherlands
HENNING WACHSMUTH, Institute of Artificial Intelligence, Leibniz University Hannover, Germany

Quality control is an, if not *the*, essential challenge in crowdsourcing. Unsatisfactory responses from crowd workers have been found to particularly result from ambiguous and incomplete task descriptions, often from inexperienced task requesters. However, creating clear task descriptions with sufficient information is a complex process for requesters in crowdsourcing marketplaces. In this paper, we investigate the extent to which requesters can be supported effectively in this process through computational techniques. To this end, we developed a tool that enables requesters to iteratively identify and correct eight common clarity flaws in their task descriptions before deployment on the platform. The tool can be used to write task descriptions from scratch or to assess and improve the clarity of prepared descriptions. It employs machine learning-based natural language processing models trained on real-world task descriptions that score a given task description for the eight clarity flaws. On this basis, the requester can iteratively revise and reassess the task description until it reaches a sufficient level of clarity. In a first user study, we let requesters create task descriptions using the tool and rate the tool's different aspects of helpfulness thereafter. We then carried out a second user study with crowd workers, as those who are confronted with such descriptions in practice, to rate the clarity of the created task descriptions. According to our results, 65% of the requesters classified the helpfulness of the information provided by the tool high or very high (only 12% as low or very low). The requesters saw some room for improvement though, for example, concerning the display of bad examples. Nevertheless, 76% of the crowd workers believe that the overall clarity of the task descriptions created by the requesters using the tool improves over the initial version. In line with this, the automatically-computed clarity scores of the edited task descriptions were generally higher than those of the initial descriptions, indicating that the tool reliably predicts the clarity of task descriptions in overall terms.

ACM Reference Format:
Zahra Nouri, Nikhil Prakash, Ujwal Gadiraju, and Henning Wachsmuth. 2023. Supporting Requesters in Writing Clear Crowdsourcing Task Descriptions Through Computational Flaw Assessment. In *28th International Conference on Intelligent User Interfaces (IUI '23), March 27–31, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3581641.3584039

1 INTRODUCTION

For more than a decade, crowdsourcing has been drawing the attention of organizations and individuals as a manner of finding solutions and earning money [18]. Crowdsourcing marketplaces aid on-demand access to an extensive range of human expertise, leading to a diverse set of cost-effective solutions and services. This thriving paradigm provides the opportunity to exploit the wisdom, abilities, and creativity of a huge pool of workers for problems that are difficult for computers but solvable using human intelligence. The general crowdsourcing process has three main steps [33]: (1) *task design*, where requesters deploy tasks (along with descriptions) on a crowdsourcing platform; (2) *task operation*, where workers take on tasks and later submit their solutions (as part of this step, they may ask questions about task details,

Manuscript submitted to ACM

Table 1. Example of two versions of the same crowdsourcing task description. The left description is the first version a user of our tool, ClarifyIt, wrote. The right description is the best refinement that the user created, according to the clarity score computed by the tool.

	First Version of Task Description	Best Version of Task Description
Title	Creation of new pieces of writing	Creation of new pieces of writing
Body	You will be responsible for writing pieces of text on arbitrary topics	You will be responsible for writing pieces of text on a variety of different topics. These topics may be selected randomly. You will be provided with a topic to write about as well as a minimum necessary word count for the piece. You should provide the writing in a typed format, which will then be submitted by email. Once your writing has been assessed, you will be compensated if you have met the criteria. If you do not meet the criteria, you will be given feedback and a further opportunity to adjust your writing and resubmit.

and requesters may give feedback); and (3) *task evaluation*, where requesters decide to accept or reject the solutions and, hence, to pay the workers or not.

The quality of solutions provided by the crowd has been the focus of much prior research on crowdsourcing [24]. Low-quality results are known as the dominant challenge in harnessing the full potential of crowdsourcing [39]. They emerge from various complications related to the three main stakeholders involved: (1) *workers* may be incompetent, novice, or unmotivated to deliver quality results [7]; (2) *requesters* may also be novices, unfair, or negligent in task design, operation, and evaluation [33], and (3) the *platform* may mediate the entire crowdsourcing process, and the requester-worker communication poorly or in a biased manner [37]. Among several factors that are known to have an impact on the quality of crowdwork, unclear task design has been emphasized as one of the most decisive factors [29]. Poor task design can lead to disappointment and frustration among workers due to a misalignment of expectations and unjustified rejection of work [13]. Eventually, it prejudices the requester-worker relationship, undermining the dynamics of crowdwork [30].

Writing clear task descriptions is thus vital for an effective task design. A task description usually combines a short title with a body containing instructions. In general, it should be easy to follow and understand, and should contain sufficient information about what workers are expected to do for the task and how it should be done [28]. Exemplarily, Table 1 shows two versions of the same task description, conveying clear differences with respect to the mentioned aspects. The quality of the instructions directly affects the workers' perception and selection of a task [36], so they have a significant influence on the workers' participation [22], task completion rate [6], and approval rate. The latter, in turn, affects their reputation and income [38] albeit putting effort and time [29]. Ultimately, task descriptions thereby impact both the final quality of results and the workers' trust and satisfaction [40]. Moreover, Khanna et al. [22] revealed that clear task descriptions enhance the usability of crowdsourcing for low-income workers.

As a matter of fact, a clear task design is of great importance for crowdsourcing processes. Unfortunately, ambiguous task descriptions have nonetheless been highlighted as a persistent challenge [4, 14, 15, 22, 33, 40]. The problem behind this is dual: First, requesters should sufficiently describe all information necessary for completing a task; however, this is often difficult without extensive crowdsourcing experience, especially for micro-tasks having a broad range of potential workers, who come from diverse cultures, have different skills, and various educational backgrounds [10]. Second,

writing clear and understandable descriptions is a challenging task in nature, due to both the subjective viewpoint of the requesters and the inherent ambiguity of natural language in general. Thus, workers may interpret the instructions they get differently [12]. Arguably, developing a tool that automatically supports task requesters in writing task descriptions with high clarity and completeness would help address the dual problem of describing all necessary information in unambiguous phrasing. To our knowledge, such a tool has not yet been published, likely because of a lack of applicable computational models that can predict the clarity of task descriptions.

In this paper, we contribute to the state of the art in crowdsourcing task design support (Section 2) by introducing *ClarifyIt*, a tool that automatically finds ambiguities and incomplete aspects of task descriptions in an iterative process. Task requesters can use the tool to improve the quality of their descriptions step by step, before deploying them on a crowdsourcing platform. The tool applies machine learning-based natural language processing methods that analyze a given task description, in order to detect common clarity flaws in the descriptions. Eventually, this can enable workers to accept a task based on improved task descriptions with more detailed information.

For the automatic task description analysis, we implemented and deployed computational models that distinguish eight predefined clarity flaws from the literature [31]: overall clarity, difficult wording, missing definitions of important terms as well as missing specifications of the desired solution, the solution format, the steps to perform, the required resources, and the criteria to meet for task acceptance (Section 3). Building on the findings of Nouri et al. [31], we developed one support vector regression model with various feature types for each task clarity flaw that predicts the degree of the flaw in a given task description (Section 4). We trained the models on the same dataset as the authors, which contains 1332 real micro-task descriptions annotated for each clarity flaw on a scale from 1 to 5 (Section 3). Our tool employs the trained regression models to score task descriptions based on the eight clarity flaws (Section 5). Requesters can use the tool to evaluate and edit their task descriptions in an arbitrary number of iterations until the scores shown by the tool reach satisfactory clarity.

We evaluated the effectiveness of the tool through two user studies, one with requesters (Section 6) and one with crowd workers (Section 7): In the first study, requesters created an initial version of their task description and then improved it using the tool as long as they considered it reasonable. We included both experienced and novice requesters, and we let the requesters partly start inside the tool directly and partly outside first. After the iterative creation of the task descriptions, all requesters completed a questionnaire asking about the helpfulness of our tool. In the second study, we asked crowd workers to compare the initial and the best-scored versions of the task descriptions from the first study. They should judge on a Likert scale from 1 to 5 which version is more understandable and complete with respect to the eight clarity dimensions. Here, workers were not informed which description was the initial version. Based on these studies, we assessed our tool following two research questions:

- **RQ1.** Based on the requesters' assessments, how helpful is the tool to identify and improve the clarity flaws in task descriptions?
- **RQ2.** Based on the crowd workers' assessments, how effectively does the tool support creating clearer task descriptions in terms of completeness and comprehensiveness?

According to our results, the tool was seen as helpful or very helpful by 65% of all requesters with respect to its general functionalities and by 62% concerning the information it provides on task description clarity (RQ1). Only 12% found the helpfulness to be low (less than 1% very low). Experienced and novice requesters similarly benefited from the tool. Interestingly, writing an initial draft outside the tool seems to be favorable. Potential improvements refer to more carefully chosen good and bad examples shown in the tool, along with more precise prediction models. In the second

study, 60%–78% of the crowd workers believed that the clarity improved concerning the different clarity dimensions in the best-scored version of the task descriptions compared to the initial version created by requesters (RQ2). This suggests that the scores computed by the tool are reliable in general. The results also indicate that the tool is most effective in improving the clarity of what workers should submit for tasks. In contrast, support for improvements with respect to simpler wording of task descriptions seems more challenging than the other clarity dimensions.[1]

2 BACKGROUND AND RELATED WORK

Text clarity, in general, has been studied in terms of readability and understandability, covering diverse aspects such as the syntax and semantics of text [23], vocabulary that causes semantic difficulties [3], the use of statistical language models for assessment [8], and more. Studies on text readability are also broadly surveyed by Kevyn [21]. Here, we summarize the literature that has investigated task clarity in crowdsourcing marketplaces in terms of (1) *tool*s and (2) *models and workflows* for task clarity improvements.

2.1 Tools for Improving Task Description Clarity

Manam and Quinn [29] developed a tool called *WingIt* for ambiguous task instructions. The tool builds on the workers' comprehension and intuition of the task requirements and the requesters' expected results. WingIt enables workers to communicate with the requester to ask for clarifications on the task ("Q&A"). In Q&A, workers offer the best answer for clarifications or directly modify the instructions ("Edit"). The requester-worker communication is either synchronous, waiting for the requester's response within three minutes, or asynchronous, submitting the result assuming the requesters will confirm the answer. Another tool of this kind is *SPROUT* [2]. SPROUT collects inquiries and utilizes recommendations from crowd workers to revise ambiguous parts of task descriptions. It supplies the requesters with inquiries and permits them to prioritize those inquiries. Such tools can help amateur workers at the cost of notable additional time on both the workers' and the requester's side. Yet, the risks of misinterpretations and wrong perceptions of workers remain, which may consequently lead to rejection and a bad reputation.

A more worker-oriented approach is taken by *Turkomatic* [26], which works based on a price-divide-solve algorithm. Turkomatic utilizes the crowd to split complex tasks and solve the sub-tasks through step-by-step guidance. It relies on qualified workers, leadership from the requesters, and a close feedback mechanism to succeed. The collaborative system *Revolt* follows a similar idea, focusing on image-labeling tasks with vague or incomplete instructions [5]. In Revolt, several workers can label an image according to the given instructions and the description written by other workers. In case of a conflict, workers relabel the image according to other workers' descriptions. Also, the Microsoft Word plug-in *Soylent* involves workers to edit, shorten, and proofread documents while hiding the complexity of task specifications [1].

CrowdForge and *Crowd4u* help decompose complex tasks written in natural language into small crowdsourcing tasks [19, 25]. However, they do not support all types of task specifications in crowdsourcing. *Fantasic* tests a task design to assist novice requesters [16]. It gathers task requirements from requesters to create and show a task description before posting on the platform, but it is also limited to a narrow set of task types. *TurKit*, finally, enables requesters to deploy tasks on MTurk iteratively [27]. Its architecture avoids obtaining redundant submissions by saving intermediate results. TurKit assumes that requesters will determine the decomposition mechanism of tasks in all cases.

[1]The Data and experiment code are available here: https://osf.io/2uqtf/?view_only=ac98b0ea6e4842fa878015d000627ccb

Table 2. Distribution of the 5-point Likert scores for each clarity flaw in the task description dataset of Nouri et al. [32]. The scores express the degree to which a clarity flaw is observable in a description, from *strongly disagree* (1) to *strongly agree* (5).

#	Statements on Clarity Flaws	1	2	3	4	5
1	I do not understand how to complete the task and what the desired solution is.	**0.48**	0.18	0.09	0.20	0.05
2	The wording is not easy to understand.	**0.42**	0.26	0.05	0.03	0.24
3	Some potentially important terms are not sufficiently defined.	**0.31**	0.28	0.07	0.10	0.24
4	The desired solution is not explained in sufficient detail.	0.24	**0.29**	0.16	0.22	0.09
5	The format in which the solution should be submitted is not sufficiently specified.	**0.32**	0.27	0.18	0.07	0.16
6	The steps to complete the task are not sufficiently defined.	0.15	**0.35**	0.24	0.09	0.16
7	Resources that are required to complete the task are not sufficiently specified.	0.18	**0.32**	0.25	0.13	0.12
8	The acceptance criteria for a solution to the task are not sufficiently specified.	**0.27**	0.21	**0.27**	0.13	0.12

In contrast to all these tools, in this work, we build models using natural language processing techniques to automatically detect ambiguities in task descriptions without workers' and platforms' involvement. Utilizing these models, we present an automated tool that supports requesters in iteratively identifying and improving clarity flaws in their task descriptions before posting on the platform. The tool does not require interaction with workers in the process, which can lead to more time and cost-efficient in obtaining clear task descriptions. This way, it avoids various challenges originating from the complications of the requester-worker communication [33].

2.2 Models and Workflows for Task Clarity

Several researchers studied influencing factors of task clarity. The effect of guidelines on the workers' awareness of task quality and, consequently, on the quality of their submissions in terms of accuracy, performance, trust, and worker satisfaction was investigated by Wu and Quinn [40]. Complementarily, Khanna et al. [22] explored the impact of user interfaces, task descriptions, and the workers' cultural background on MTurk workers with low digital capabilities. They suggested simplifying the task descriptions and localizing language to leverage the usability of workers. Similarly, the influence of an uncomplicated task design on workers' motivation was studied by Finnerty et al. [11], providing evidence that clear instructions increase their awareness and focus, leading to higher-quality results.

For complex writing tasks, Salehi et al. [35] suggested a workflow that begins with workers posting their questions about the task, then discussing them with the requester, and writing a draft. The requester then votes on the drafts, and the workers revise the paragraphs based on the rankings and submit the final paragraph. This workflow is costly in terms of time and money, and it greatly depends on the requester-worker relationship as well as a high-quality feedback mechanism, which seems hard to ensure. Likewise, *TaskMate* relies on workers to enhance the clarity of the task descriptions [28]. It suggests workers identify the unclarities of a task description in the form of questions and offer multiple reasonable answers for each question, other workers rate the best answer, which clarifies the ambiguities, and the workers perform on the improved task description. This approach puts all the responsibility for improving the clarity of task descriptions on workers and assumes that they collaborate well. Hence, its effectiveness depends on the workers' quality.

In *Daemo*, requesters deploy several instances of their task on the platform and receive feedback from workers to improve their description [15]. While this method proved effective in principle, its dependence on the subjective judgments of a restricted number of workers does not fit well for large crowds with diverse backgrounds and skills. Besides, the pilot step is costly in terms of time and money. To avoid such issues, Gadiraju et al. [14] developed a

computational model that emphasizes predictive features, such as role clarity and goal clarity, as the significant aspects of task clarity. It remains open, though, to how to best operationalize the model to improve description clarity.

In contrast, the computational approach presented in this paper suggests a workflow to work towards clarity iteratively. It relies on the requester's performance throughout the process, avoiding diverse complications discussed in [33]: from low-quality submissions by workers to difficulties of a poor requester-worker relationship and the improper feedback system in the process. Indirectly, we still consider the workers' opinions; our computational models are built on annotations of clarity flaws in real-world task descriptions. Thereby, we facilitate the development of an assistant tool to improve requesters' task descriptions clarity, which in turn is expected to improve workers' comprehension and submission quality. The tool helps inexperienced requesters realize what information is essential for creating complete and clear task descriptions.

3 TASK DESCRIPTION CLARITY

In this section, we briefly describe task description clarity in crowdsourcing concerning both comprehensibility and completeness, summarizing the task clarity dimensions discussed by Nouri et al. [31] and encoded in their dataset. Below, we use the dataset to train the models that compute the degree of clarity flaws in a given task description.

3.1 Task Description Clarity

Task description clarity pertains to the dual principal quality of textual descriptions that determines the extent to which all required information is provided to obtain optimal solutions as well as the intelligibility degree of the instructions written by requesters in natural language for a massive network of crowd workers from diverse backgrounds.

In practice, inexperienced requesters often write unclear task descriptions, partly due to a narrow perception of the diversity of the potential task participants in terms of demographics, competence, and similar. Moreover, inexperienced requesters may also be unaware of the importance of a high-quality task design and its substantial effect on the quality of the solutions submitted by workers. Unclear task descriptions can cause imprecise or incorrect submissions contrary to the task requesters' expectations, leading to task rejection and distrust between requesters and workers.

Nouri et al. [31] collected eight common clarity flaws from prior research on task descriptions and their various dimensions in terms of completeness or clear phrasing. We shortly discuss them here, as they provide the basis for our computational model and, in turn, our tool:

(1) *Overall clarity.* The description is not comprehensible and/or lacks information about how to complete the task.
(2) *Wording and phrasing.* The words and/or grammar used in the description are not intelligible.
(3) *Definition of important terms.* Some keywords to correctly understand the tasks are not defined sufficiently.
(4) *Specification of desired solution.* The solution expected from workers is not clarified in adequate detail.
(5) *Specification of desired format.* The expected format of the solution to submit is not clarified sufficiently.
(6) *Specification of steps.* The steps workers should take to submit solutions are not clarified sufficiently.
(7) *Specification of required resources.* Resources required to solve the task are missing, such as tools, links, or data.
(8) *Statement of acceptance criteria.* The requirements for accepting a submission are not clarified sufficiently.

3.2 Data for Studying Task Description Clarity

The defined clarity flaws served as the basis for annotation guidelines that Nouri et al. [31] used to create a corpus. The corpus consists of 1332 real-world task descriptions initially published on Amazon Mechanical Turk (MTurk) from

October 2013 to September 2014. Each task description contains the title text, a dot (as a separator), and the body of the task. Crowd workers annotated according to the eight statements expressing the existence of the defined clarity flaws in the given task descriptions. The annotation task was deployed on the MTurk platform, and workers were asked to rate the extent on a 5-point Likert scale to which they agreed with each statement for the given description.

Table 2 summarizes the statements on task clarity flaws as well as the distribution of final Likert scores for all the task descriptions for each clarity flaw. In total, the 1332 descriptions span 31,027 tokens.

4 COMPUTATIONAL ASSESSMENT OF FLAWS

The tool we present below aims to aid requesters of crowdsourcing tasks in identifying clarity flaws in their task descriptions and in improving the descriptions in an iterative manner. This section describes the computational models we created to assess clarity flaws automatically in detail. We built on findings of Nouri et al. [31] who evaluated the effectiveness of two different types of models in clarifying flaw classification: a feature-based support vector machine (SVM) [20] and transformed model based on BERT [9]. The authors observed no consistent improvements in clarity flaw detection using BERT models; rather, the SVM performed better overall in this specific use case while being much less resource-intensive. Therefore, we rely on similar feature-based methods here, too, but we developed models that numerically quantify clarity flaws.

4.1 Features for Modeling Clarity Flaws

Nouri et al. [31] studied the feasibility of the automatic classification of task description clarity by applying natural language processing techniques to the description's plain text only. To this end, they proposed six types of features for learning to classify that we adopt for our purposes. This set of feature types was elaborately collected for feature-based modeling techniques, and the work's findings indicated that the classifiers could assess almost all the clarity flaws in task descriptions. Therefore, we rely on the features for our feature-based models. We shortly summarize the features here, but we refer to the original paper for more details:

(1) *Content.* TF-IDF (term frequency–inverse document frequency) scores of all lower-cased token 1- to 3-grams.
(2) *Length.* 26 normalized length features, such as the number of words and characters per sentence, the number of punctuation marks per sentence, and similar.
(3) *Style.* Part-of-speech 1- to 3-grams, phrase 1- to 3-grams, characters 3-grams, and the 100 most frequent lower-cased words in the training data.
(4) *Subjectivity.* Scores for the subjectivity, polarity, negativity, positivity, and objectivity of task descriptions computed by Textblob library.
(5) *Readability.* Flesch-Kincaid Grade Level, Coleman-Liau, ARI, Flesch Reading-Ease, Gunning-Fog Index, LIX, SMOG Index, RIX, and Dale-Chall Index metrics.
(6) *Flaw-specific.* Eight task-specific features, four of which count web-related terms, URLs, specified time intervals, and defined rewards. The others model the distribution of named entities, part-of-speech categories, words often appearing in clear/unclear texts, and complex words.

4.2 Regression Models for Flaw Assessment

Our tool integrates computational models to predict the degree to which each flaw is presented in a given task description. Unlike Nouri et al. [31], who classified flaws, we, therefore, employ supervised *regression* to obtain numerical scores

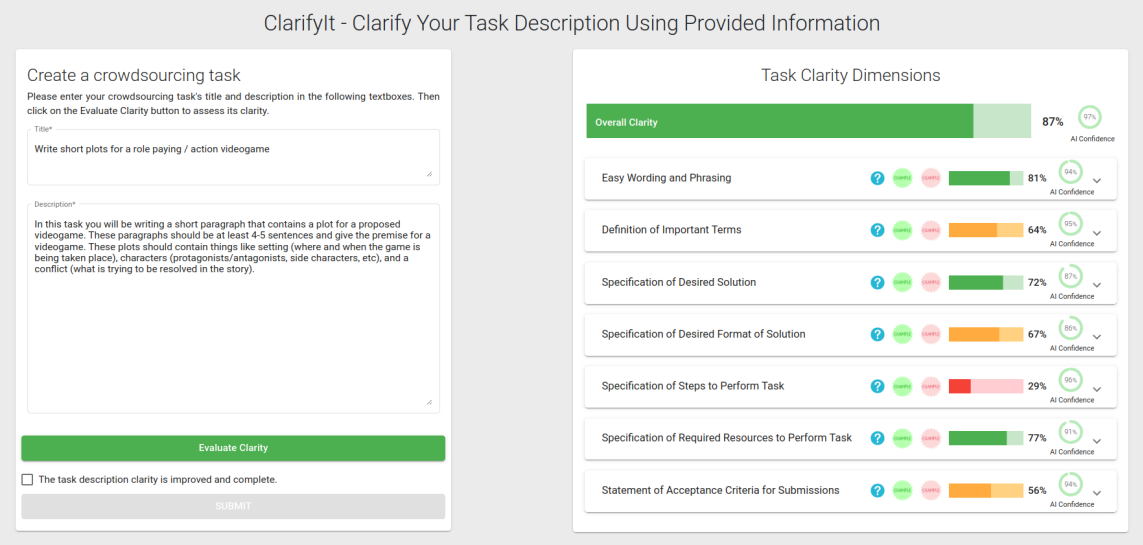

Fig. 1. The user interface of our writing assistance tool *ClarifyIt*: On the left, the requester enters the title and description of a crowdsourcing task. Once *Evaluate Clarity* is clicked on, the tool automatically assesses the task's Clarity and gives feedback on various clarity dimensions on the right. The requester can then improve and repeat the process until the description is clear.

representing degree. Since we aim to study how to help requesters improve the clarity of the task descriptions rather than what the best regression method is, we decided to follow the authors' findings on the different methods discussed above. In particular, we employed support vector regression (SVR), which is known to be one of the best methods for feature-based regression. Given the complete dataset from Section 3, we trained one separate SVR model for each of the eight clarity flaws annotated in the dataset. We point out, though, that further improvements in flaw assessment may be possible in future work, for instance, by employing recent transformer models, such as DeBERTa [17].

To select the ideal set of features for each regressor, we used the *SelectKBest* class from *scikit-learn* [34] which scores all features and keeps only the k highest scoring features for some defined k. For each clarity flaw, we tested *SelectKBest* on SVR with 20 different cost hyperparameters (in the range: 2^i for $-10 <= i <= 10$) and 15 different values of k (in the range: $100 * i$ for $1 <= i <= 15$). Among the total 300 different models for each dimension, we selected the hyperparameters corresponding to the best-performing model in terms of mean squared error, computed by 5-fold cross-validation. Eventually, the feature sets and the corresponding optimized hyperparameters were used to train the models for each dimension.

5 CLARIFYIT: A WRITING ASSISTANCE TOOL FOR TASK DESCRIPTIONS

Using the developed computational models, we created a tool (called *ClarifyIt* where 'It' refers both to the task description and the Iterative process) used to assist requesters of crowdsourcing tasks in iteratively writing clear task descriptions. This section describes its user interface, and the intended process of working with it.

5.1 Architecture

ClarifyIt is a web-based tool using a three-layered architecture. In particular, the architecture consists of (a) the *presentation layer* (frontend) providing a user interface through which crowdsourcing requesters interact with the system; (b) the *application layer* (backend) handling the computation of clarity scores as well as logging and similar; and (c) the *data layer*, which stores the pre-trained models and logs. The tool is implemented using HTML/CSS and Angular on the frontend and Python on the backend. The code is available on GitHub [2].

5.2 User Interface

The user interface, as shown in Figure 1, broadly contains two sections: (a) the *input section* on the left through which requesters can feed their task description to the system; and (b) the *evaluation section* on the right shows feedback on task clarity dimensions, their corresponding scores, and the scores' confidence values on a scale from 0 to 100. For each dimension, the evaluation section also includes a brief description, an example of a good task description concerning that dimension, and an example of a bad one. The description and examples can be accessed by clicking on the respective icons: [3], [4].

5.3 Process

The requester can either draft a task description (consisting of a title and a body) from scratch within the tool or copy it from external sources. Upon clicking the button *Evaluate Clarity*, the presentation layer sends the task description to the application layer for computing clarity scores. In the application layer, the task description is passed through all feature-type modules to compute their corresponding feature values. These feature values are fed into the pre-trained regression models, fetched from the data layer, corresponding to each dimension. Then, a dimension score, in terms of a percentage value, is computed for each dimension by scaling the score predicted by the corresponding model accordingly. On the other hand, the standard deviation of the top three performing pre-trained models' predictions is scaled to compute each dimension's confidence score. Finally, the dimension and confidence scores are sent to the presentation layer. Ultimately, the requester can then decide to consider the predictions to either improve the clarity of their available task description or to create it step-by-step utilizing repeating the sketched process in the tool.

6 EVALUATION WITH TASK REQUESTERS

Before we look at the quality of the task description resulting from our tool, ClarifyIt, we designed an experiment to study the influence of ClarifyIt on requesters and workers in relation to task instructions. Figure 2 illustrates the experimental setup where we first evaluate the tool's helpfulness with requesters that write crowdsourcing task descriptions (cf. ❶). Through the user study, we assess the requesters' view of how well the tool assists them by providing the utilities and required information to create a clear task description. We also analyze how effectively the tool improves task description clarity according to the computational models it employs. In the following, we detail the user study where we instructed a set of requesters to create a task description iteratively using ClarifyIt and then complete a questionnaire based on their experience.

[2] https://github.com/Nix07/clarifyIt
[3] Question Mark (https://icons8.com/icon/80684/question-mark) icon by https://icons8.com
[4] Example (https://icons8.com/icon/kP5VLEsdwqY8/example) icon by https://icons8.com

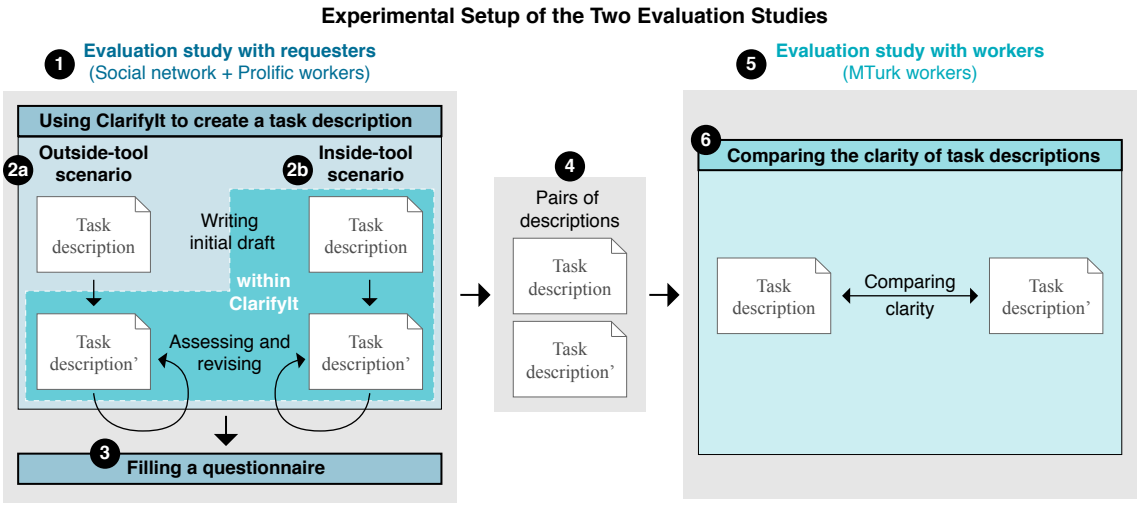

Fig. 2. An overview on the experimental setup of the study that evaluates the helpfulness and effectiveness of ClarifyIt with task requesters in creating clear task descriptions and workers in better understanding the task.

6.1 Experimental Setup

In the user study, we relied on the following setup:

6.1.1 Task. To avoid forcing the participants to deal with a specific domain they may not know much about, we defined six scenarios somewhat abstractly as creating a crowdsourcing task. We randomly assigned a scenario to participants and asked them to imagine themselves as a requester of that task scenario in mind, create a description, and improve it using ClarifyIt. One of the given scenarios is the following:

> *'Imagine a situation where you have an entity like a set of images, objects, audio files, or similar to be annotated by crowd workers according to some conditions. Write down a task description explaining the task to crowd workers.'*

6.1.2 Participants. As requesters, we recruited (a) researchers from our social network and (b) crowd workers from *Prolific* (cf. 1). 122 participants completed our study, of which 14 were researchers from our social network. The prolific workers were required to have English as their first language, an approval rate higher than 95%, and at least 100 previous task submissions. Based on a pilot study, we calculated 15 minutes to finish the task and paid £ 2.50 (£ 7.50 per hour, as recommended by Prolific). The participants from our network did their work voluntarily.

6.1.3 Experiments. After signing up and getting instructions, participants had to give their consent to participate on the landing page. They were then randomly assigned to either of the following two settings, designed to investigate the impact of the information provided by our tool on the clarity of the initial task description created by requesters:

(1) *Outside-tool scenario.* Here, participants had to create an initial task description according to the given scenario before entering the tool. Then, they should copy it into the tool to check and refine it there (cf. 2a).
(2) *Inside-tool scenario.* Here, participants had to enter the tool directly. They saw the scenario there and then had to create, check, and refine the description in the tool (cf. 2b).

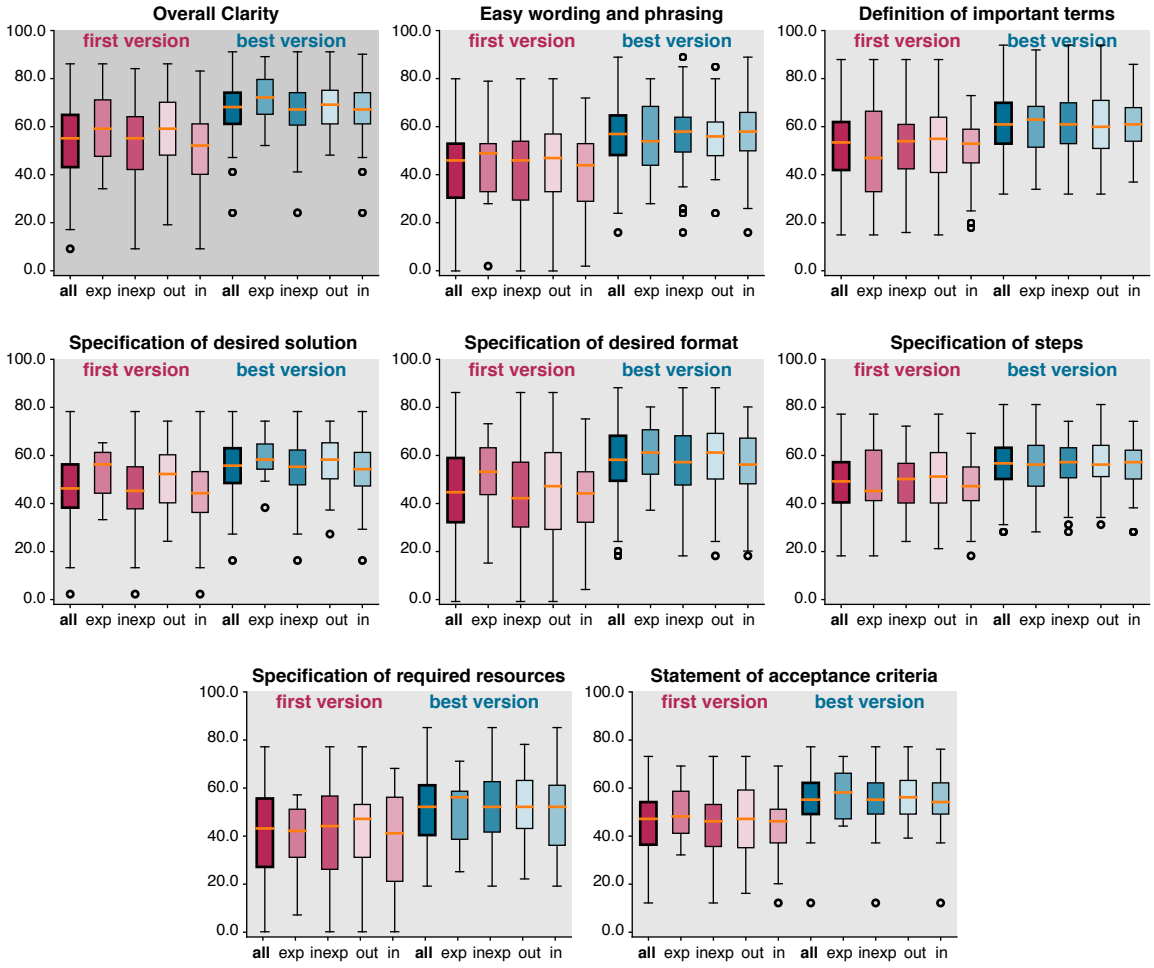

Fig. 3. Evaluation with task requesters: The scores of all eight considered clarity dimensions for the *first version* of the requesters' task description as well the *best version*, manually created and automatically scored using our tool, ClarifyIt: The box-and-whiskers plots show the results of *all* participants, *exp*erienced vs. *inexp*erienced participants as well as *out*side-tool scenario vs. *in*side-tool scenario participants. In all cases, the scores improved notably from the first to the best version.

After creating the initial task description, participants could evaluate and improve the clarity of their description based on the clarity dimension scores and other information provided by the tool. The process of assessing and improving the description clarity could be done in iterations until the task description reaches sufficient clarity—according to the clarity scores shown by the tool or otherwise by the requester's judgment. Next, the participants were to answer a questionnaire (cf. ❸) containing 17 questions about their experience with crowdsourcing in general (such as their years of experience and the platforms they know) and the tool in the study. In the end, requesters could write comments and suggestions for improving the tool's usability. In the following, we discuss the result of the user study with requesters.

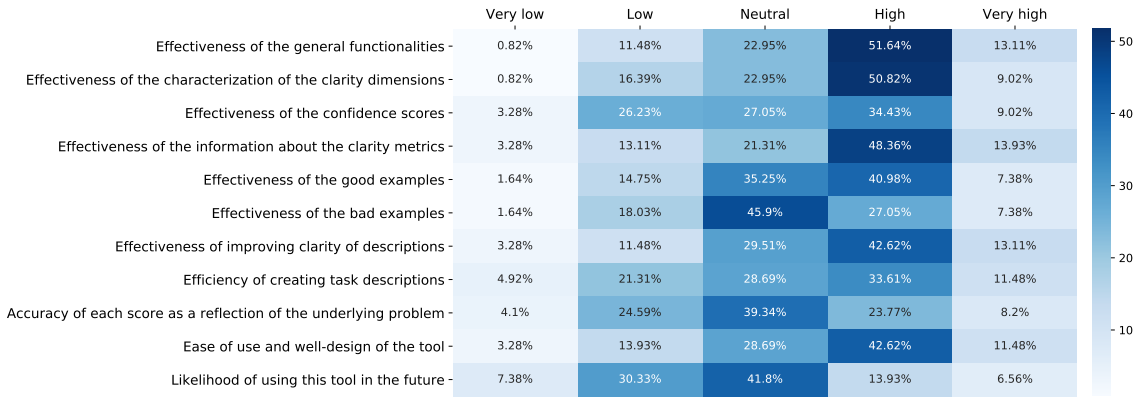

Fig. 4. Evaluation with task requesters: Distribution of the scores given to the questions in the questionnaire about the experience with our tool, ClarifyIt. For most questions, most requesters saw the tool's effectiveness as high.

6.2 Results

In total, 122 participants with up to 13 years of experience requesting tasks on crowdsourcing platforms completed our study. We call those 107 with no prior experience *novice* and the other 15 *experienced* requesters. They mainly knew Amazon Mechanical Turk, Toloka, and Prolific. Among the completed submissions, 57 came from the outside-tool scenario and 65 from the inside-tool scenario.

Although we did not instruct participants to do so, 24 of those assigned to the outside-tool scenario (42%) revised their task description right after viewing the information about the clarity dimensions provided on the tool before evaluating the clarity. This led to an average overall clarity improvement of eight percentage points, implying that the information provided by the tool about how to write a clear task description is effective from the beginning.

6.2.1 Task Descriptions. Figure 3 shows that, on average, participants improved the clarity of their task descriptions using ClarifyIt. The best-scored version notably improves over the initial version on all eight dimension scores. All differences are significant at $p < 0.01$ according to a paired t-test. Furthermore, they all have either a medium or large effect size. For instance, the difference between the best-scored version (M = 64.61, SD = 10.87) and initial version (M = 50.75, SD = 16.20) of the overall clarity dimension has a *Cohen's d* value of 1.01, indicating a large effect size.

Figure 3 also illustrates that the inexperienced participants write a clearer definition of essential terms and the required steps to perform the task in the first version. However, the initial description's overall clarity, wording, desired solution, and format created by experienced participants are scored higher by our tool. This observation indicates that experienced requesters sometimes overlook the importance of explaining new keywords that may confuse workers from outside the domain. Besides, performing the task may sound vital to the task creator. Consequently, they miss providing necessary information on how the workers should do the task and submit their results.

We also observe that the first version of the descriptions written by participants who saw the clarity dimensions through the inside-tool scenario has no higher clarity score than the initial descriptions created through the outside-tool scenario. We can interpret that the general knowledge of clarity aspects of task descriptions does not influence the descriptions' clarity. Yet, the score of each clarity aspect for a given description can help the writer improve the clarity. Altogether, the results also show that the clarity of the best-scored descriptions improved using ClarifyIt in all cases.

Although participants generally improved their task description within iterations, 24% of the iterations decreased the dimension scores. This indicates that changes to task descriptions during those iterations harmed their clarity (as judged by the tool). Further, we found that the best task description version of 42 participants (34%) differs from the final version based on the overall clarity score. This raises the need for an undo functionality in ClarifyIt so that requesters can revert to the previous version of their task description when the score drops after an update. A few participants requested the same through open-ended comments. For example, one requester suggested:

> R1: *"I would like to see the previous score in order to get an idea about the archived improvements."*

6.2.2 User Experience. In light of RQ1, the requesters' responses to the questionnaire in Figure 4) suggest that the most useful feature of ClarifyIt are the *general functionalities* with about 65% being positive about it (13.11% very high, 51.64% for high) and only 12% negative (0.82% very low, 11.48% low). They are followed by the information provided about the *clarity metrics* with 62%, and the *characterization of the clarity dimensions* with 60% of the participants' votes. Participants also gave positive feedback regarding the overall helpfulness of ClarifyIt in the open-ended comments. For example, one requester described the usefulness of clarity dimensions as follows:

> R2: *"The tool was extremely helpful. I lacked imagination in creating the task specifics however the metrics were genuinely helpful in clarifying what workers would need to know."*

Besides, 56% of the participants believed that the tool is *easy to use and well-designed*, and 54% expressed that it is highly helpful in *improving the clarity of task descriptions*. One of the requesters commented:

> R3: *"I could see through each of my edits how I was making the description clearer and easier to understand."*

The *good examples* provided for clarity dimensions and the accuracy of the scores for the unclarities are rather beneficial, with only 16% and 38% of negative opinions, respectively. The latter is probably due to the distributional shift between the training's task descriptions and the participants'. For example, one requester reported that the tool did not recognize the task description content corresponding to a few dimensions:

> R4: *"Naming the categories helped by making it clear which aspects should be present in the description. Unfortunately, however, the AI did not recognize when I included the aspects I had previously forgotten in the description."*

The *bad examples* provided for unclarity dimensions were neither seen as helpful nor useless on average, with the majority voting for neutral (46%). We also discovered that participants provided varied responses to the *efficiency* of the process of creating clear task descriptions, with 29% negative, 39% neutral, and 32% positive opinions.

In total, 34% of the requesters wanted to know what specific changes they could make to their task description to improve clarity. Three of them read as follows:

> R5: *"[It] should tell you which parts to improve"*

> R6: *"It does not provide suggestions as to where it could be improved for example where you may benefit from adding a comma."*

> R7: *"Would be more helpful if it could show suggestions of better wording to improve it."*

Most voters (42%) could not anticipate whether they would use the tool in the future, and 38% expressed that they would probably not seek assistance to create their task descriptions. Enhancements in good or bad examples, improvements in models' performance, and dynamic task-specific suggestions for clarity improvements can increase the tool's popularity and helpfulness for users. We plan to work on these ideas in future work.

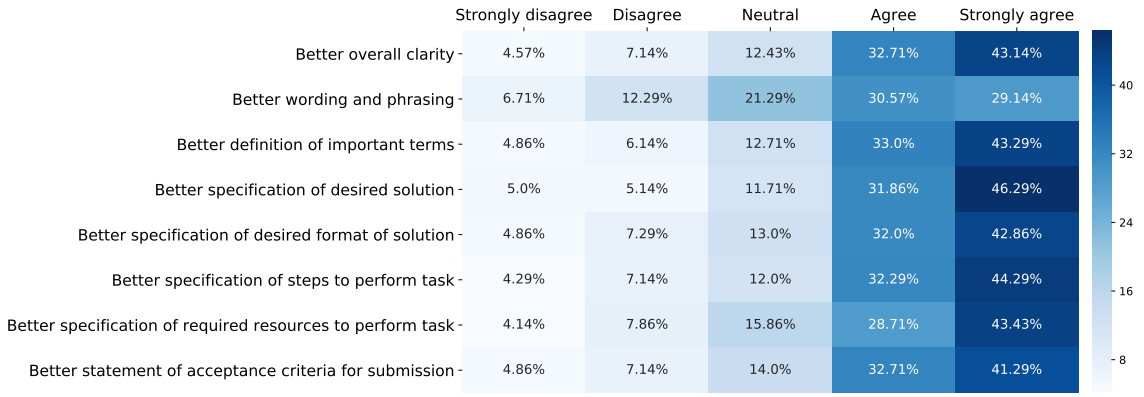

Fig. 5. Evaluation with crowd workers: Distribution of the scores on improvements in terms of the eight clarity flaws of the best-scored versions of the 100 task descriptions over the initial version. Workers did not know which version is which one.

7 EVALUATION WITH CROWD WORKERS

Given the user study results with requesters, we carried out a second user study with crowd workers to evaluate the effectiveness of our tool, ClarifyIt, in improving the task description clarity from the workers' perspective (cf. ❺). Concretely, we asked the workers to compare the initial version and the best version of the task descriptions created by the requesters (cf. ❻) in terms of the eight clarity dimensions considered to judge whether requesters managed to create clearer task descriptions using the tool.

7.1 Experimental Setup

In this study, we used the following setup:

7.1.1 Data. We randomly sampled 100 pairs of task descriptions created in the user study with requesters (cf. ❹). Each pair included the initial version that a requester wrote and the best version from subsequent iterations in terms of the overall clarity score computed by the respective model.

7.1.2 Participants. The task descriptions in the dataset of Nouri et al. [31] have been initially published on Amazon Mechanical Turk (MTurk), and they have also been annotated for clarity flaws by MTurk workers. Therefore, we also decided to employ MTurk workers to assess differences in the given task descriptions' clarity.

We considered only workers from the US, Canada, UK, Ireland, Australia, South Africa, and New Zealand for language proficiency reasons. To participate in our study, they needed at least 10,000 approved submissions on MTurk, and an approval rate of a minimum 98%. Aligned with our budget constraints, we employed seven workers to vote on the clarity improvement of each description pair and paid each worker US-$ 1.25 for an estimated time of six minutes. While a higher number of participants would further increase statistical reliability, seven votes seem enough to identify general tendencies. To increase the quality of results, we accepted only submissions from workers who passed the two attention checks discussed below.

Table 3. Evaluation with crowd workers: (a) Average agreement of the seven workers on the 100 task description pairs, and (b) proportion of task descriptions whose clarity improved over the initial version by using our tool, ClarifyIt, according to the workers; both for each clarity dimension considering neutral votes either for (*case 1*) or against (*case 2*) improvements.

		(a) Average agreement		**(b) Improved descriptions**	
#	**Clarity Dimensions**	**Case 1**	**Case 2**	**Case 1**	**Case 2**
1	Overall clarity	81%	89%	86%	98%
2	Wording and phrasing	69%	81%	68%	98%
3	Important terms	81%	90%	86%	98%
4	Desired solution for task	82%	91%	88%	98%
5	Desired format of solution	78%	89%	89%	97%
6	Steps to perform task	80%	89%	88%	98%
7	Required resources to perform task	79%	89%	82%	98%
8	Acceptance criteria for submission	78%	90%	88%	97%

7.1.3 Experiments. In accordance with the statements in Table 2, each of the eight statements expressed that Task Description 1 is more clear than Task description 2 in terms of the respective clarity dimension. The workers were asked to vote to what extent they agreed with the statement on a 5-point Likert scale from *strongly disagree* to *strongly agree*.

The first attention check tested whether the workers read the statements carefully through two objective statements: (a) *"Task description 1' is longer than 'Task description 2"* and (b) *'There are more words in 'Task description 1"*. Given the four pairs per task, we used (a) for the fourth statement in tasks #1 and #3, and (b) in #2 and #4. If the statement was true for a pair, workers with *strongly agree* or *agree* passed the attention check; if it was false, those with *strongly disagree* or *disagree*. Only workers who passed all four checks were considered for the second part.

For the second attention check, we copied one arbitrary statement of each comparison task and repeated it for its last statement to test whether the workers expressed their opinion carefully. For example, the third statement *'The potentially important terms are better defined.'* was used as the tenth statement of Task #1 again. To pass this attention check, we required workers to consistently express whether they (strongly) agree or (strongly) disagree with the statement in both occurrences for the given comparison. Due to the subjective nature of the statement, we considered that workers might change their opinion slightly. Hence, answering *neutral* was also accepted, leading to passing the attention check and accepting and paying for their submissions.

7.2 Results

We acquired 700 submissions with votes from 92 different workers on improvements in the 100 pairs of task descriptions. To obtain uniform votes, we automatically reversed the votes for those comparison tasks where Task Description 1 was set to the initial version, meaning workers voted against improvements in the best-scored version.

7.2.1 Task Descriptions. In light of RQ2, Figure 5 shows the distribution of scores from all submissions. According to workers' votes, the best-score version of the task descriptions is clearer than their initial version concerning all clarity dimensions. The most significant improvements were in *specification of desired solution* (78%, 46.29% strongly agree and 31.86% agree), *specification of steps to perform task* (77%), *overall clarity* and *definition of important terms* (both 76%) dimensions based on the judgment of all voters. Besides, 60%–75% of voters saw improvements in other clarity dimensions. However, improvements in *wording and phrasing* dimension received the most negative votes (19%, 6.71% strongly disagree and 12.29% disagree) as well as most neutral votes (21%).

7.2.2 Agreement. We classified the workers' votes into binary labels (positive or negative), representing votes for and against clarity improvements, respectively. We computed the majority vote for the agreement among the seven voters for each description pair in two ways, interpreting the neutral votes as against (*Case 1*) and as for (*Case 2*) clarity improvements in the best-scored version of the task description.

Table 3a shows the average agreement for all eight dimensions, and Table 3b the proportion of improved task descriptions for clarity dimensions in both cases. In Case 1, the agreement among the voters is in the range of 68% (for *better wording and phrasing*) to 82% (for *desired format better specified*), and the improvement level in the task descriptions clarity is in the range of 68% to 89% (for the same dimensions). In Case 2, the ranges change to 81% (for *better wording and phrasing*) to 91% (for *desired format better specified*), and the improvement level increases to 97%–98% across all clarity dimensions. Concerning RQ2, we conclude that the view of the workers clearly indicates the impact of our tool in creating clear task descriptions.

8 CONCLUSIONS AND FUTURE WORK

Unclear task descriptions were written by requesters of tasks, often to low-quality results, since they easily cause misunderstanding and misinterpretation of the tasks. Previous work identified such unclarity as one of the primary issues limiting success in crowdsourcing. In this paper, we have studied the impact of a tool called *ClarifyIt* that we developed to support requesters in creating and revising task descriptions in an iterative process until a sufficient level of clarity is reached. The tool employs machine learning-based natural language processing techniques to detect eight common clarity flaws in task descriptions automatically. Its workflow does not require worker intervention, making it potentially more efficient and effective than prior solutions. We evaluated the effectiveness based on the requesters' and workers' opinions from two user studies. In the first study, requesters used our tool to write task descriptions and to improve their clarity. We then let the requesters assess how helpful the tool is in improving clarity (RQ1). In the second study, the crowd workers judged the quality of initial and improved task descriptions to test whether the requesters improved their task description clarity using our tool (RQ2).

In light of RQ1, the first study's results indicate that the tool's primary functionality and provided information are particularly helpful. Moreover, the requesters saw the tool as well-designed and effective in identifying and improving a description's clarity. In light of RQ2, the crowd workers' judgments suggest that all clarity flaws of task descriptions created through ClarifyIt notably improved on average. Here, the clarity of the wording and phrasing in the task descriptions is the most challenging dimension to predict computationally and, thus, to assist requesters.

However, different points remain to be considered. First, the effectiveness of the computational models in detecting the clarity flaws in task descriptions can likely be improved in different aspects. Particularly, a larger dataset containing a broader set of descriptions of micro-tasks annotated for clarity flaws may allow for training more precise models. In this regard, identifying the wording clarity in task descriptions seems to be the most challenging. Improving the prediction of this dimension may need refined models capturing the vagueness in task descriptions. Regarding the effectiveness of the developed tool, some requesters noted that they would like to receive real-time feedback on clarity while typing a task description. To make the process more efficient and effective, others aimed to get suggestions for clarity improvements in their descriptions (similar to *Grammarly*) as well as an undo button that enables them to restore previous versions of a task description in case of reaching lower clarity scores after changes. Still, we are convinced that crowdsourcing platforms could benefit already from integrating our ClarifyIt, for example, as a plug-in tool to provide their requesters with automated assistance in writing task descriptions.

Due to the influence of various decisive factors of effective task design, such as fair estimation of time and payment and a well-designed feedback system, as well as clarity of task instructions on the quality of submission by workers, investigating the effectiveness of ClarifyIt on the final results involves out-of-scope factors influencing the study result. Therefore, we here relied on insights discussed in [40] to point out whether our work serves as a practical approach to increasing the quality of workers' final results as a high-level goal. Wu and Quinn [40] discussed that the clarity of task instructions influences the workers' behavior and that requesters must be knowledgeable about the task's requirements and the principles of task description design. The result of the evaluation study on our tool also supports that ClarifyIt effectively assists requesters in understanding and mitigating the clarity flaws of their task instructions, which is essential in addressing the challenge of low-quality submissions by workers.

Despite some room for improvement, we conclude that the developed tool has the potential to impact real-world crowdsourcing task descriptions in practice positively. In the future, it will be helpful to evaluate the latest transformer-based models, such as DeBERTA [17], for computational assessment to study whether such models may further improve clarity flaw assessment in crowdsourcing task descriptions. It is also worth investigating how improvements in important dimensions of an effective task design influence the quality of final results. Finally, we think that similar approaches to providing automated support for textual content creators could be explored in other domains. In principle, respective tools may assist the creators in identifying and improving their text clarity flaws according to any operationalizable clarity specification.

REFERENCES

[1] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 313–322.

[2] Jonathan Bragg, Daniel S Weld, et al. 2018. Sprout: Crowd-powered task design for crowdsourcing. In *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, 165–176.

[3] Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability revisited: The new Dale-Chall readability formula.* Brookline Books.

[4] Jesse Chandler, Gabriele Paolacci, and Pam Mueller. 2013. *Risks and Rewards of Crowdsourcing Marketplaces.* Springer New York, New York, NY, 377–392. https://doi.org/10.1007/978-1-4614-8806-4_30

[5] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2334–2346.

[6] Jenny J Chen, Natala J Menezes, Adam D Bradley, and T North. 2011. Opportunities for crowdsourcing research on amazon mechanical turk. *Interfaces* 5, 3 (2011), 1.

[7] Chun-Wei Chiang, Anna Kasunic, and Saiph Savage. 2018. Crowd coach: Peer coaching for crowd workers' skill growth. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–17.

[8] Kevyn Collins-Thompson and James P Callan. 2004. A language modeling approach to predicting reading difficulty. In *Proceedings of the human language technology conference of the North American chapter of the association for computational linguistics: HLT-NAACL 2004*. 193–200.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[10] Djellel Difallah, Elena Filatova, and Panos Ipeirotis. 2018. Demographics and dynamics of mechanical turk workers. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 135–143.

[11] Ailbhe Finnerty, Pavel Kucherbaev, Stefano Tranquillini, and Gregorio Convertino. 2013. Keep it simple: Reward and task design in crowdsourcing. In *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI*. 1–4.

[12] Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. 2011. CrowdDB: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, 61–72.

[13] Ujwal Gadiraju and Gianluca Demartini. 2019. Understanding worker moods and reactions to rejection in crowdsourcing. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media*. 211–220.

[14] Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. 2017. Clarity is a worthwhile quality: On the role of task clarity in microtask crowdsourcing. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*. ACM, 5–14.

[15] Snehalkumar (Neil) S Gaikwad, Mark E Whiting, Dilrukshi Gamage, Catherine A Mullings, Dinesh Majeti, Shirish Goyal, Aaron Gilbee, Nalin Chhibber, Adam Ginzberg, Angela Richmond-Fuller, et al. 2017. The daemo crowdsourcing marketplace. In *Companion of the 2017 ACM conference on computer supported cooperative work and social computing*. 1–4.

[16] Philipp Gutheim and Björn Hartmann. 2012. Fantasktic: Improving quality of results for novice crowdsourcing users. *EECS Dept., Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2012* 112 (2012).

[17] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *CoRR* abs/2111.09543 (2021). arXiv:2111.09543 https://arxiv.org/abs/2111.09543

[18] Jeff Howe. 2006. The rise of crowdsourcing. *Wired magazine* 14, 6 (2006), 1–4.

[19] Kosetsu Ikeda, Atsuyuki Morishima, Habibur Rahman, Senjuti Basu Roy, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. 2016. Collaborative crowdsourcing with crowd4u. *Proceedings of the VLDB Endowment* 9, 13 (2016), 1497–1500.

[20] Thorsten Joachims. 1998. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, Claire Nédellec and Céline Rouveirol (Eds.). Springer Verlag, Heidelberg, DE, Chemnitz, DE, 137–142. /brokenurl#joachims98.ps

[21] Collins-Thompson Kevyn. 2014. Computational assessment of text readability. *ITL-International Journal of Applied Linguistics* 165, 2 (2014), 97–135.

[22] Shashank Khanna, Aishwarya Ratan, James Davis, and William Thies. 2010. Evaluating and improving the usability of Mechanical Turk for low-income workers in India. In *Proceedings of the first ACM symposium on computing for development*. 1–10.

[23] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.* Technical Report. Naval Technical Training Command Millington TN Research Branch.

[24] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*. 1301–1318.

[25] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 43–52.

[26] Anand P Kulkarni, Matthew Can, and Bjoern Hartmann. 2011. Turkomatic: automatic recursive task and workflow design for mechanical turk. In *CHI'11 extended abstracts on human factors in computing systems*. 2053–2058.

[27] Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. 2010. Turkit: human computation algorithms on mechanical turk. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 57–66.

[28] VK Chaithanya Manam, Dwarakanath Jampani, Mariam Zaim, Meng-Han Wu, and Alexander J. Quinn. 2019. TaskMate: A Mechanism to Improve the Quality of Instructions in Crowdsourcing. In *Companion Proceedings of The 2019 World Wide Web Conference*. 1121–1130.

[29] VK Chaithanya Manam and Alexander J Quinn. 2018. Wingit: Efficient refinement of unclear task instructions. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*.

[30] Brian McInnis, Dan Cosley, Chaebong Nam, and Gilly Leshed. 2016. Taking a HIT: Designing around rejection, mistrust, risk, and workers' experiences in Amazon Mechanical Turk. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 2271–2282.

[31] Zahra Nouri, Ujwal Gadiraju, Gregor Engels, and Henning Wachsmuth. 2021. What is unclear? computational assessment of task clarity in crowdsourcing. In *Proceedings of the 32nd ACM Conference on Hypertext and Social Media*. 165–175.

[32] Zahra Nouri, Ujwal Gadiraju, Gregor Engels, and Henning Wachsmuth. 2021. What is Unclear? Computational Assessment of Task Clarity in Crowdsourcing. In *Submitted for publication*.

[33] Zahra Nouri, Henning Wachsmuth, and Gregor Engels. 2020. Mining Crowdsourcing Problems from Discussion Forums of Workers. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 6264–6276. https://doi.org/10.18653/v1/2020.coling-main.551

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[35] Niloufar Salehi, Jaime Teevan, Shamsi Iqbal, and Ece Kamar. 2017. Communicating context to the crowd for complex writing tasks. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. 1890–1901.

[36] Thimo Schulze, Stefan Seedorf, David Geiger, Nicolas Kaufmann, and Martin Schader. 2011. Exploring task properties in crowdsourcing–An empirical study on Mechanical Turk. (2011).

[37] David Schwartz. 2018. Embedded in the crowd: Creative freelancers, crowdsourced work, and occupational community. *Work and Occupations* 45, 3 (2018), 247–282. https://doi.org/10.1177/0730888418762263 arXiv:https://doi.org/10.1177/0730888418762263

[38] M Six Silberman, Joel Ross, Lilly Irani, and Bill Tomlinson. 2010. Sellers' problems in human computation markets. In *Proceedings of the acm sigkdd workshop on human computation*. 18–21.

[39] Daniel S Weld, Christopher H Lin, and Jonathan Bragg. 2015. Artificial intelligence and collective intelligence. *Handbook of Collective Intelligence* (2015), 89–114.

[40] Meng-Han Wu and Alexander James Quinn. 2017. Confusing the crowd: Task instruction quality on amazon mechanical turk. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*.