

# Practice of Efficient Data Collection via Crowdsourcing at Large-Scale

Alexey Drutsa, Viktoriya Farafonova, Valentina Fedorova,  
Olga Megorskaya, Evfrosiniya Zerminova, Olga Zhilinskaya\*

August 08, 2019

## Abstract

Modern machine learning algorithms need large datasets to be trained. Crowdsourcing has become a popular approach to label large datasets in a shorter time as well as at a lower cost comparing to that needed for a limited number of experts. However, as crowdsourcing performers are non-professional and vary in levels of expertise, such labels are much noisier than those obtained from experts. For this reason, in order to collect good quality data within a limited budget special techniques such as incremental relabelling, aggregation and pricing need to be used. We make an introduction to data labeling via public crowdsourcing marketplaces and present key components of efficient label collection. We show how to choose one of real label collection tasks, experiment with selecting settings for the labelling process, and launch label collection project at Yandex.Toloka, one of the largest crowdsourcing marketplace. The projects will be run on real crowds. We also present main algorithms for aggregation, incremental relabelling, and pricing in crowdsourcing. In particular, we, first, discuss how to connect these three components to build an efficient label collection process; and, second, share rich industrial experiences of applying these algorithms and constructing large-scale label collection pipelines (emphasizing best practices and common pitfalls).

## 1 Introduction

Modern machine learning algorithms require a large amount of labelled data to be trained. Crowdsourcing has become a popular source of such data due to its lower cost, higher speed, and diversity of opinions comparing to labelling data with experts. However, performers at crowdsourcing marketplaces are non-professional and their labels are much noisier than that of experts [20]. For this reason, to obtain good quality labels via crowdsourcing and under a limited budget, special methods for label collection and processing are needed. The goal of this tutorial is to teach participants how to efficiently use crowdsourcing marketplaces for labelling data.

Crowdsourcing platforms can process a wide range of tasks (a.k.a., human intelligence tasks, HITs), for instance: information assessment (e.g., used in ranking of search results); content categorization (e.g., used in text and media moderation, data cleaning and filtering); content annotation (e.g., used in metadata tagging); pairwise comparison (e.g., used in offline evaluation, media duplication check); object segmentation, including 3D (e.g., used in image recognition for self-driving car); audio and video transcription (e.g., used in speech recognition for voice-controlled virtual assistant); field surveys (e.g., used to verify business information and office hours); etc. Two examples of tasks are in Figure 1.

---

\* Affiliation: Yandex (16, Leo Tolstoy St., Moscow, Russia, 119021; [www.yandex.com](http://www.yandex.com)).

## Example: binary classification



## Examples: object segmentation

Figure 1: Examples of human intelligence tasks (HITs) that can be executed on crowdsourcing platforms: binary classification (the left side) and object segmentation (the right side).

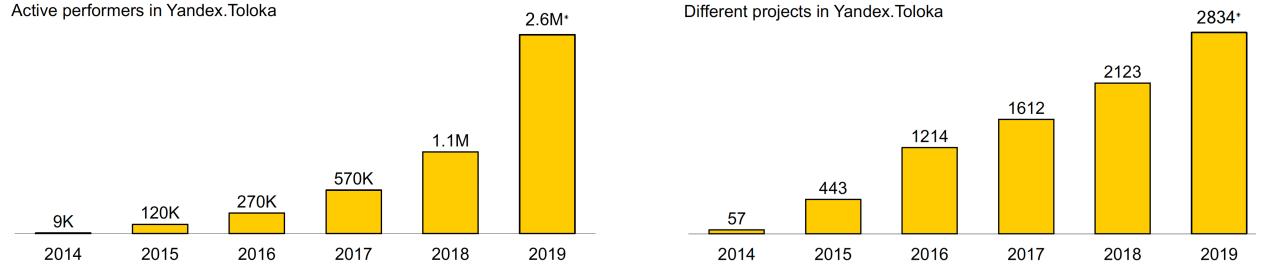


Figure 2: Crowdsourcing growth: Yandex experience (\* statistic for 2019 is obtained via an extrapolation based on the first 7 months of 2019).

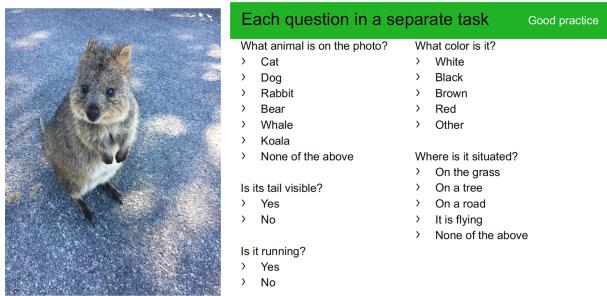
Crowdsourcing is widely used in modern industrial IT companies in permanent manner and on a large scale. The development of their products and services strongly depends on the quality and costs of labelled data. For instance, Yandex’s crowdsourcing experience is presented in Figure 2, where the substantial growth is seen in terms of both active performers and projects. Currently, 25K performers execute around 6M HITs in more than 500 different projects everyday at Yandex.Toloka<sup>1</sup>.

## 2 Key components for efficient data collection

We discuss *key components* required to collect labelled data: proper decomposition of tasks (construction of a pipeline of several small tasks instead of one large human intelligent task), easy to read and follow task instructions, easy to use task interfaces, quality control techniques, an overview of aggregation methods, and pricing. Quality control techniques include: approaches “before” task execution (selection of executors, education and exam tasks), the ones “within” task execution (golden sets, motivation of performers, tricks to remove bots and cheaters), and approaches “after” task execution (post verification/acceptance, consensus between performers). We share best practices, including: pitfalls when designing instructions & interfaces, important settings in different types of templates, training and examination for performers selection, important aspects in tasks instructions for performers, pipelines for evaluating the process of labelling.

<sup>1</sup><https://toloka.yandex.com/for-requesters>

## Case of decomposition: a lot of questions



## Instruction ambiguity for a rare case: example

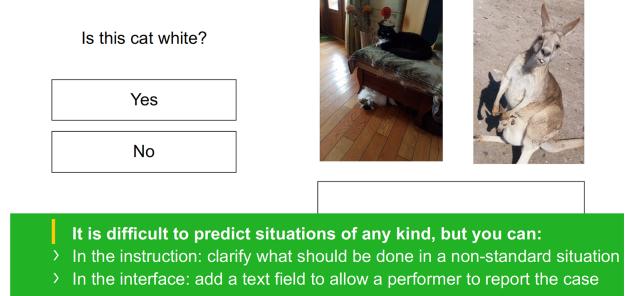


Figure 3: The left side: an example of a single task with multiple questions. The right side: an example of rare cases that should be taken into account when building task interfaces and instructions.

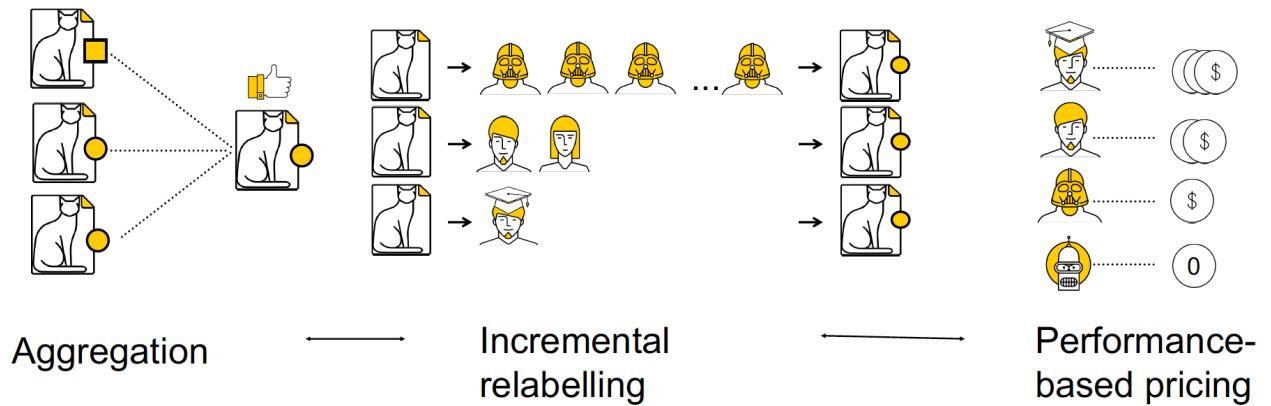


Figure 4: Interconnection between three approaches that make crowdsourcing more efficient: aggregation, incremental relabelling (IRL), and performance-based pricing.

Figure 3 (left side) contains an example of a single task with multiple questions. In this case, the best practice is to split this task into several one such that each question will be in a separate HIT. Figure 3 (right side) contains few example images for a binary classification task where a performer should decide whether a cat is white or not. These examples are rare cases that should be taken into account when building task interfaces and instructions.

### 3 Efficiency methods: aggregation, IRL, and pricing

The next approaches are the main ones that make crowdsourcing more efficient:

- *Methods for aggregation in crowdsourcing.* Classical models: Majority Vote, Dawid-Skene [4], GLAD [25], Minimax Entropy [28]. Analysis of aggregation performance and difficulties in comparing aggregation models in unsupervised setting [19, 9]. Advanced works on aggregation: combination of aggregation and learning a classifier [15], using features of tasks and performers for aggregation [16, 24, 11], ensemble of aggregation models [7], aggregation of crowdsourced pairwise comparisons [2].
  - *Incremental relabelling (IRL).* Motivation and the problem of incremental relabelling: IRL based on Majority Vote; IRL methods with performers quality scores [10, 6, 1]; active learn-

	MV	DS	GLAD	MME
Categories (K)	○ ○ ○	△ ○ □	○ ○ ○	△ ○ □
Objects (J)	□ □ □	□ □ □	cat dog milk	cat dog milk
Workers (W)	Darth Vader	Robot Head	Graduation Cap	Robot Head
Number of parameters	0	$WK^2 + K$	$W + J$	$(W + J)K^2$

Figure 5: Summary on the key properties of the main aggregation methods: Majority Vote, Dawid-Skene [4], GLAD [25], Minimax Entropy [28].

ing [13]. Connections between aggregation and IRL algorithms. Experimental results of using IRL at crowdsourcing marketplaces.

- *Pricing of tasks in crowdsourcing.* Practical approaches for task pricing [8, 23, 3, 26]. Theoretical background for pricing mechanisms in crowdsourcing: efficiency, stability, incentive compatibility, etc. Pricing experiments and industrial experience of using pricing at crowdsourcing platforms.

## 4 Crowdsourcing pipeline to highlight objects on images

Attendees of our practice session create and run a crowdsourcing pipeline for a real problem on real performers. We propose to highlight objects of a certain type on images. A set of photos of real roads is taken as an example (since such a task is vital for autonomous vehicle development). Participants should select a type of objects to be highlighted: e.g., people, transport, road, curb, traffic lights, traffic signs, sidewalk, pedestrian crossing, etc. Highlighting of objects of the selected type is proposed to be done by means of bounding boxes via a public crowdsourcing platform. The formal setup of our task is as follows:

- each object of a selected type
- in each photo from the dataset
- needs to be highlighted by a rectangle (bounding box).

For instance, if traffic signs are chosen, then Figure 6 demonstrates how a photo should be processed. Participants propose their crowdsourcing pipelines and compare them with ours. For the described task, we suggest to use the pipeline depicted in Figure 7 as the baseline.

This simple pipeline consists of three projects. The tasks for the first one are binary classification HITs. The second project contains HITs with a bounding box highlighting tool. The third project

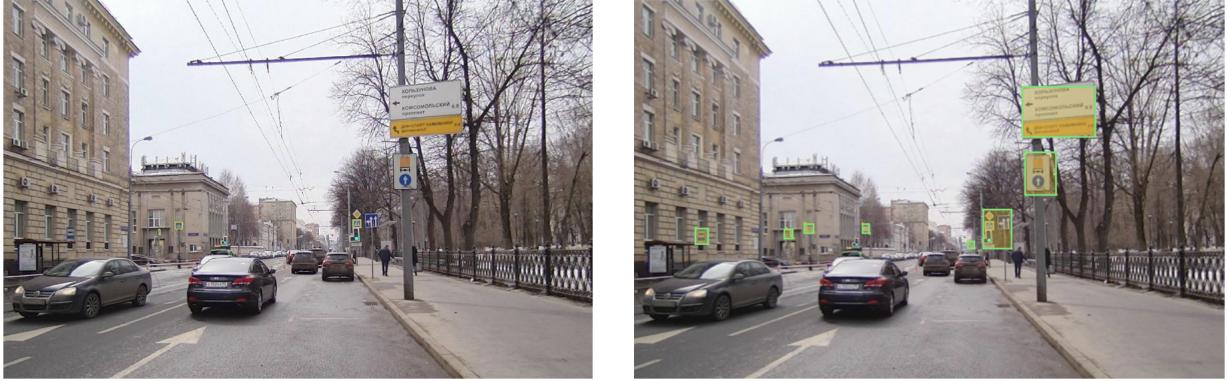


Figure 6: An example of a photo before (the left side) and after processing (the right side): all traffic signs are highlighted by bounding boxes.

### Suggested pipeline

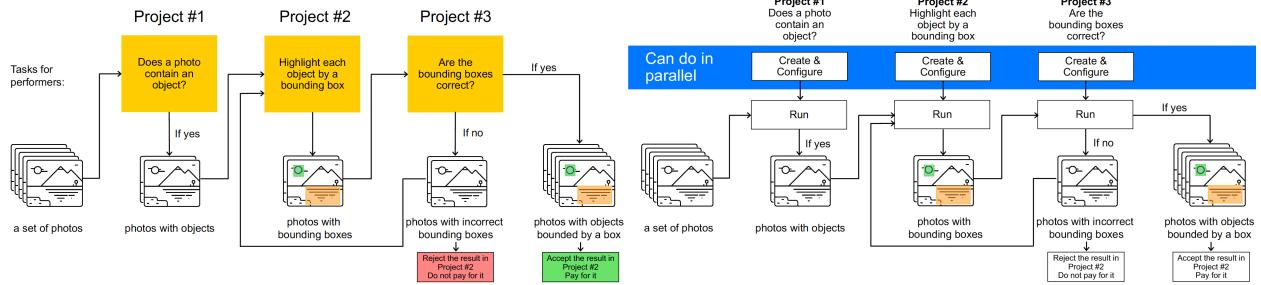


Figure 7: The left side: the suggested crowdsourcing pipeline to solve the problem of object highlighting on photos. The right side: how to work on creation and running of the suggested pipeline.

designed to verify the results obtained from the second project. The summary on these projects is shown in Figure 8.

Attendees of our practice session create, configure, and run this pipeline on real crowd performers. We run this pipeline to process 100 images and highlight traffic signs. Our results are as follows.

1. Project #1: "Does a photo contain traffic signs?"
  - 100 photos evaluated
  - within 4 min on real performers
  - cost: \$0.3 + Toloka fee
2. Project #2: "Highlight each traffic sign by a bounding box"
  - 67 photos processed
  - within 5.5 min on real performers
  - cost: \$0.67 + Toloka fee
3. Project #3: "Are traffic signs highlighted by the bounding boxes correctly?"
  - 90 photos evaluated
  - within 5 min on real performers
  - cost: \$0.36 + Toloka fee



Figure 8: Short descriptions of HITs of three types used in the suggested crowdsourcing pipeline.

## 5 Related tutorials

Previous tutorials consider different components of labeling process separately and did not include practice sessions. On the contrast, the goals of our tutorial is to explain the main algorithms for incremental relabelling, aggregation, and pricing and their connections to each other, and to teach participants the main principles for setting up an efficient process of labeling data at a crowdsourcing marketplace. Following is a summary of relevant topics covered in previous tutorials:

- “Crowdsourcing: Beyond Label Generation” presented at NIPS 2016, ALC 2017, and KDD 2017. A part of this tutorial devoted to an overview of empirical results about performers reaction to pricing.
- “Crowd-Powered Data Mining” conducted at KDD 2018. The introduction and the first part of this tutorial was devoted to the standard process of crowdsourcing label collection and aggregation.
- “Crowdsourced Data Management: Overview and Challenges” was held at SIGMOD’17 and partly focused on methods for aggregating crowdsourced data.
- “Truth Discovery and Crowdsourcing Aggregation: A Unified Perspective” was conducted at VLDB 2015 and dedicated to methods for aggregating crowdsourced data.
- “Spatial Crowdsourcing: Challenges, Techniques, and Applications” was conducted at VLDB 2016. This tutorial focused on using crowdsourcing for spatial tasks and included efficient methods for task targeting, aggregating data, and the effect of pricing for such tasks. Our tutorial will be devoted to another type of crowdsourcing tasks which is multiclassification.

## Tutorial materials

The tutorial materials (slides and instructions) are available at <https://research.yandex.com/tutorials/crowd/kdd-2019>.

## References

- [1] Ittai Abraham, Omar Alonso, Vasilis Kandylas, Rajesh Patel, Steven Shelford, and Aleksandrs Slivkins. How many workers to ask?: Adaptive exploration for collecting high quality labels. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 473–482, 2016.
- [2] X. Chen, P. N Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of WSDM*, 2013.

- [3] Justin Cheng, Jaime Teevan, and Michael S Bernstein. Measuring crowdsourcing effort with error-time curves. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1365–1374. ACM, 2015.
- [4] A. P. Dawid and A. M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.
- [5] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, and Philippe Cudré-Mauroux. Scaling-up the crowd: Micro-task pricing schemes for worker retention and latency improvement. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.
- [6] Seyda Ertekin, Haym Hirsh, and Cynthia Rudin. Learning to predict the wisdom of crowds. *arXiv preprint arXiv:1204.3611*, 2012.
- [7] Siamak Faridani and Georg Buscher. Labelboost: An ensemble model for ground truth inference using boosted trees. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.
- [8] Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. Incentivizing high quality crowdwork. In *Proceedings of the 24th International Conference on World Wide Web*, pages 419–429. International World Wide Web Conferences Steering Committee, 2015.
- [9] Hideaki Imamura, Issei Sato, and Masashi Sugiyama. Analysis of minimax error rate for crowdsourcing and its application to worker clustering model. *arXiv preprint arXiv:1802.04551*, 2018.
- [10] P G Ipeirotis, F Provost, V S Sheng, and J Wang. Repeated labeling using multiple noisy labelers. In *Data Mining and Knowledge Discovery*, pages 402–441. Springer, 2014.
- [11] Yuan Jin, Mark Carman, Dongwoo Kim, and Lexing Xie. Leveraging side information to improve label quality control in crowd-sourcing. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*, 2017.
- [12] H. Kim and Z. Ghahramani. Bayesian classifier combination. In *International conference on artificial intelligence and statistics*, pages 619–627, 2012.
- [13] Christopher H Lin, M Mausam, and Daniel S Weld. To re(label), or not to re(label). In *Second AAAI conference on human computation and crowdsourcing*. AAAI, 2014.
- [14] Chao Liu and Yi-Min Wang. Truelabel+ confusions: A spectrum of probabilistic models in analyzing multiple ratings. *arXiv preprint arXiv:1206.4606*, 2012.
- [15] V. C Raykar, S. Yu, L. H Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *The Journal of Machine Learning Research*, 11:1297–1322, 2010.
- [16] P. Ruvolo, J. Whitehill, and J. R Movellan. Exploiting commonality and interaction effects in crowdsourcing tasks using latent factor models. 2013.
- [17] Nihar Shah and Dengyong Zhou. No oops, you wont do it again: Mechanisms for self-correction in crowdsourcing. In *International conference on machine learning*, pages 1–10, 2016.
- [18] Nihar Shah, Dengyong Zhou, and Yuval Peres. Approval voting and incentives in crowdsourcing. In *International Conference on Machine Learning*, pages 10–19, 2015.

- [19] Aashish Sheshadri and Matthew Lease. Square: A benchmark for research on computing crowd consensus. In *First AAAI conference on human computation and crowdsourcing*, 2013.
- [20] R. Snow, B. O’Connor, D. Jurafsky, and A. Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008.
- [21] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 155–164, 2014.
- [22] Jeroen Vuurens, Arjen P de Vries, and Carsten Eickhoff. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR’11)*, pages 21–26, 2011.
- [23] Jing Wang, Panagiotis G Ipeirotis, and Foster Provost. Quality-based pricing for crowdsourced workers. 2013.
- [24] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010.
- [25] J. Whitehill, T. Wu, J. Bergsma, J. R Movellan, and P. L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.
- [26] Ming Yin, Yiling Chen, and Yu-An Sun. The effects of performance-contingent financial incentives in online labor markets. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.
- [27] Liyue Zhao, Gita Sukthankar, and Rahul Sukthankar. Incremental relabeling for active learning with noisy crowdsourced annotations. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 728–733. IEEE, 2011.
- [28] D. Zhou, Q. Liu, J. C Platt, C. Meek, and N. B Shah. Regularized minimax conditional entropy for crowdsourcing. *arXiv preprint arXiv:1503.07240*, 2015.

## A Introduction to the requester interface

*Interface for requesters* is discussed on the example of the crowdsourcing marketplace Yandex.Toloka. This include key concepts and definitions: projects and task instructions, templates for projects, pools of tasks, task suites, honeypots, quality control, performer skills, tasks with post acceptance and auto acceptance. Project creation includes quick start and main settings for labelling data. Types of project templates are multiple choice task to classify items, side by side comparisons, surveys to collect opinions on a certain topic, audio transcription, voice recording, object selection to locate one or more objects in an image, spatial task to visit a certain place and perform a simple activity.

Key types of instances in Yandex.Toloka are a project, a pool, and a task. A project defines the structure of tasks and how to perform them. A requester configures a task interface, a task

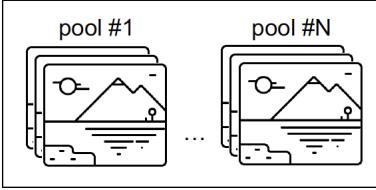
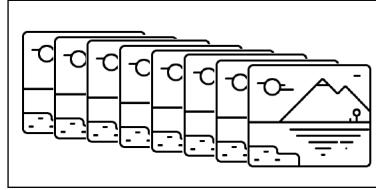
Project	Pool	Task
<ul style="list-style-type: none"> <li>&gt; Defines the structure of tasks</li> <li>&gt; Defines how to perform them</li> </ul>	<ul style="list-style-type: none"> <li>&gt; Is a batch of tasks</li> <li>&gt; Defines access of performers</li> </ul>	<ul style="list-style-type: none"> <li>&gt; A particular input data</li> <li>&gt; Results for it from performers</li> </ul>
		
<b>Configure in a project:</b> <ul style="list-style-type: none"> <li>&gt; Input and output data types</li> <li>&gt; Task interface</li> <li>&gt; Task instruction</li> </ul>	<b>Configure in a pool:</b> <ul style="list-style-type: none"> <li>&gt; Performer filters</li> <li>&gt; Quality control mechanisms</li> <li>&gt; Overlap settings</li> </ul>	

Figure 9: Key types of instances in Yandex.Toloka: a project, a pool, and a task.

instruction, input and output data types in a project. A pool is a batch of tasks and defines access of performers. A requester configures performer filters, quality control mechanisms, overlap settings, and pricing in a pool. A task is a particular input data and results for it from performers.