Yandex Toloka

SIGMOD'2020 hands-on tutorial

Practice of Efficient Data Collection via Crowdsourcing: Aggregation, Incremental Relabelling, and Pricing

Setting up and running label collection projects **Instruction**

Yandex Toloka for requesters: https://toloka.ai/

For dataset: http://tlk.s3.yandex.net/relsubstitutes/dataset_Y.tsv where Y is any number from 1 to 10

Tutorial slides: https://research.yandex.com/tutorials/crowd/sigmod-2020

Table of contents

Suggested pipeline	3
Project creation. Main steps	4
Key types of instances in Yandex.Toloka	4
Project #1 Does the photo contain an item?	5
Project creation	6
Pool creation	9
Preparing and uploading a file with tasks	11
Receiving responses	14
Project #2 Find a similar item in an online store	15
Project creation	16
Pool creation	
Preparing and uploading a file with tasks	
Project #3 Does the item found look similar to the initial one?	24
Project creation (similar to the 1 st project)	25
Pool creation	
Preparing and uploading a file with tasks	
Receiving responses	
Upload reviewed results	
Review assignments in the interface (another way of results validation)	
Project #4 Which item is more similar?	
Project creation	
Pool creation	
Preparing and uploading a file with tasks	
Receiving responses	
Appendix: Expanded code of the projects	46

Suggested pipeline



Project creation. Main steps



Key types of instances in Yandex. Toloka

Project	Pool	Task

- > Defines the structure of tasks
- > Defines how to perform them



- Configure in a project:
- > Input and output data types
- > Task interface
- > Task instruction

Is a batch of tasks

>

Defines access of performers



Configure in a pool:

- > Performer filters
- > Quality control mechanisms
- > Overlap settings

A particular input data

>

Results for it from performers



Project #1 Does the photo contain an item?



Important: Before you start using Toloka, make sure that the **English** language is selected.

Yandex Toloka	Projects Users Skills	Profile Messages	3 \$0.00 / \$5.59 ₩ Toloka requester
Projects Active V	Only mine V		Kind project + Create project
Title 💠	Private comment \$	Created \$	Open pools
		No data	

Project creation

1. Click the button + Create project

Yandex Toloka	Projects Users Skills Profile Messages		Toloka requester
Projects Active ~	Yandex ~	Find project	+ Create project
Title 💠	Private comment \$	Created 💠	Open pools

2. Choose the **Image classification** template.

Templates	
Use templates to create and publish tasks that meet your needs. You can use a template as-is or adjust it to your input data and response format.	
Classification	
Video moderation Users watch a video and choose one of the options. The task has a video player and several radio buttons.	Image classification Good for image classification and tagging. The template includes an image and several radio buttons.
Select O Preview	Select • Preview

3. Enter a clear project name and description. Important: It will be visible for real people.

PROJECT NAME	Are there shoes in the picture?	\times			
DESCRIPTION	Look at the picture and tell whether there are shoes in it.	\times			
NSTRUCTIONS		•			
INSTRUCTIONS	Look at the photo and decide, whether there are shoes in the photo.				
	If yes, click "YES" If no, click "NO"				
	For example, there are shoes in the photo. Therefore, the correct answer is "YES"				

4. Write short and simple instructions. To include an image in the instruction just paste the link from the dataset provided by pressing button.



- 5. Define parameters for the input and output data:
 - The "**image**" input data field with the link type will be used to pass the image links to the performers.

You will be able to upload the file with links to the pool later.

- The "result" field will be used to receive performer's responses.
- The "like" field in the template is used to pass the response to the question "Do you like the photo?". Our project doesn't require this checkbox, so you don't need an output field for it. Let's remove it.

Input data	o	utput data	<	>
image (URL)	•	result (string)	•	
		like (boolean)	DELETE.	

- 6. Create the task interface in the HTML block.
 - Delete the line with the checkbox component: {{field type="checkbox" name="like" label="Do you like the photo?" hotkey="q"}}
 - Add a question: does the image include a certain object? Example: <div>Are there shoes in the picture?<div>
 - Replace "label" with your response options (change "Good" to "Yes" and "Bad" to "No"). Example:

```
{{img src=image width="100%" height="400px"}}
 Are there <b>shoes</b> in the picture?
{{field type="radio" name="result" value="Yes" label="Yes"}}
{{field type="radio" name="result" value="No" label="No" }}
{{field type="radio" name="result" value="404" label="Picture
not found" }}
```

- 7. Leave JavaScript and CSS block unchanged.
- 8. Click the **Preview** button to see the performer's view of the task. You will see tasks with standard pictures on the page. You will set the number of tasks per page when configuring a pool.
- 9. Select the radio buttons in the preview and make sure that the task can be completed.



10. Click Save button to save the project.

Note. To edit project parameters, click the button in the list of projects or **Project** actions \rightarrow Edit on the project page.

re there shoes in the p	icture? — act	tive				Project actions	5 ^
ivate comment						Edit	
ne object - (post-verification) - попытка 1						Archive	
atistics for 7 days						Preview	
ibmitted tasks Spent	Quality: control tasks	Quality: training tasks	Average submit time U	sers Banne	ed users		
0.36 \$	-	-	23 s 2	6 0			
pols Training Statistics Qu	ality control						
bols Training Statistics Qu	ality control Filters					Add a p	oool
ools Training Statistics Qu Search Title \$	ality control Filters Priority \$	Progress	Status +		Started 0	Add a p To be complet	pool ted
bools Training Statistics Qu Search Title ¢ Are there shoes in the picture?	ality control Filters Priority + 0	Progress 36 / 36	Status € ✓ Complete	d January 21	Started 0 , 2020 9:47 PM	Add a p To be complet	pool ted

Pool creation

- 1. Click Add pool.
- 2. Give the pool any convenient name and description. You are the only one who can see them.
- 3. Specify the pool parameters:
 - Set the price per task page (for example, \$0.01).

	Price per task s	suite			
	You can add one or r	more tasl	ks to the page. Enter the tot	tal price for all tasks on the	
PRICE IN US DOLLARS 🕜	0.01	×	FEE 🕜	0.005	
	+ Dynamic pricing]			

- 4. Set up user filters.
 - Select English-speaking performers using the Language = English filter.

	Performers	Copy settings from
	Filter performers who can access the task. Toloka has users from different countries, so don't forget to filter by language and regio	n. Learn more
ADULT CONTENT 📀	No	
	Add filter	✓ Create skill
	PROFILE Languages	sh 🛛 🗙 🗎 +

5. Set up <u>quality control</u>: <u>Control tasks</u>. Ban performers who give incorrect responses to control tasks. Example:

Add r All rul	rules to get more accurate responses. Iles work independently.	
NON-AUTOMATIC ACCEPTANCE	No REVIEW PERIOD IN DAYS	
APTCHA FREQUENCY 🕜 NO	one ~	
CONTR	ROL TASKS	÷.
	If number of responses v ≥ 3 × +	
	If number of responses > ≥ 3 × and correct responses (%) <	
	If number of responses v ≥ 3 × + and correct responses (%) v < 60 × = then ban v on project v	
	If number of responses v ≥ 3 × + and correct responses (%) v < 60 × = then ban v on project v 10 × days v	

This rule will be triggered when the performer completes 3 control tasks in the pool. If the performer gives at least 3 responses to the control tasks and the percentage of correct responses is less than 60%, they lose access to the project for 10 days. If the percentage of correct responses is over 60%, the performer can pass to the next task page. The rule will be triggered after the next control task.

Optionally, add other quality control rules.

6. Overlap. This is the number of users who will complete the same task. For example, 3.

		Overlap	
		Specify how many	performers you want to complete each task in the pool.
	OVERLAP 📀	3	×
DYNAM	IIC OVERLAP 🕜	Off	

7. Optionally, specify the percentage of top-rated performers in the <u>Speed / Quality ratio</u>. Important: This can slow down pool completion.

Speed/quality ratio 💿									
Top %	Online	Time							
Specify the	percentage of	top-rated act	ive users	who car	n access	tasks in	the poo	ol.	
3523 🔶 Speed	All 90%	80% 70%	60%	50%	40%	30%	20%	' I 10%	352 ∱ Quality
60% top-rate The task is a	ed performers wailable to 21 1	were selecte I 3 active user	i. 5.						

8. Time allowed for completing a task page (for example, 300 seconds).

	Parameters		
TIME FOR COMPLETING A TASK PAGE IN SECONDS.	300 ×	POOL CLOSING DATE 📀	2021-06-07
KEEP TASK ORDER 📀	No	IE BEFORE POOL CLOSES IN SECONDS	0
	PO	OL PRIORITY IN PROJECT 👔	0

9. Save the pool.

Preparing and uploading a file with tasks

- 1. Download TSV-file with images by link that you were provided at the beginning of the practice session.
- 2. Upload pool tasks from this file. Important: If you changed the name of the input field, change it in the file as well

Projects → Are there shoes in the picture? → Are there shoes in this picture?								
Are there shoes	in this picture? — closed	b	Statistics	▲ Download results	~ E	Edit	× 0	
POOL TASKS (File example for task uploading (t	sv, UTF-8)) 🔘		0	0/				
1 Upload			U	%				
0 task suites	0 training task		D	one 0				
0 tasks	0 control task	0					0	

Select **Smart mixing** and specify the number of tasks per page. For example: 9 main tasks and 1 control task.

Settings for file upload 💿					
Tasks per page			Ado	ling tasks to pool (tasks	s_1_project_test1)
				TASKS	FOR POOL
By empty row Se	t manually	Smart mixing		100 tasks	0 training tasks
Main tasks	9	×		0 golden tasks	
Training tasks	0				
Control tasks Show advanced settings	1	×			Cancel Add
File example for task uploading		Close Upload			

3. Create control tasks.

Note. Control tasks are tasks with the correct response known in advance. They are used to track the performer's quality of responses. The performer's response is compared to the response you provided. If they match, it means the performer answered correctly.

• Click Edit \rightarrow Create control tasks.

test pool - clos	sed	Statistics	▲ Download results	×E	dit	•
POOL TASKS (File example for task uploading (tsv, UTF-8)) 🔘					
Upload files Delete	Edit	0	%			
0 task suites	0 training task	C	lone 0			
100 tasks	0 control task	0				0

Edit tasks						
Use main tasks as a starting point to create control tasks or training tasks.						
Control tasks are for checking the quality of responses from performers. They contain correct responses to compare with actual responses.						
Training tasks are for	teaching performer	s how to complete tasks. They cor	tain correct responses and hints			
earn more						
Create control tasks	Create training tas	ks	Download			
Create control tasks	Create training tas	ks Responses from performers 💠	Download			
Create control tasks ID \$ 1975eb01	Create training tas Overlap ¢ 3	ks Responses from performers \$	Download Last updated 07/26/2019 22:18:06			
Create control tasks ID \$ 1975eb01 1975eabe	Create training tas Overlap ¢ 3 3	ks Responses from performers \$ 0 0	Download Last updated + 07/26/2019 22:18:06 07/26/2019 22:18:06			

• Check the "result" output field that is used to match the user response to the control one, select the response and click **Save and go to next**.



Enter correct responses for your control tasks. In small pools, control tasks should account for 10-20% of all tasks.

Tip. Make sure to include different variations of correct responses in equal amounts. Open the **Control tasks** \rightarrow **Distribution of correct responses for control tasks** tab.

Projects → Does the image contains t → test pool → Uploaded tasks						
Edit tasks						
Use main tasks as a s	tarting point to crea	ate control tasks or training tasks.				
Control tasks are for compare with actual	checking the quality responses.	y of responses from performers. They	y contain correct responses to			
Training tasks are for	teaching performer	rs how to complete tasks. They conta	in correct responses and hints.			
Learn more						
Learn more Main 90 Control t Create control tasks	asks 10 Training Create training tas	g tasks 0	Download			
Learn more Main 90 Control t Create control tasks	asks 10 Training Create training tas Overlap \$	g tasks 0 sks Responses from performers +	Download Last updated •			
Learn more Main 90 Control t Create control tasks ID ¢ 19f9102c	asks 10 Training Create training tas Overlap ¢ 3	g tasks 0 sks Responses from performers * 0	Download Last updated 07/26/2019 21:45:27			

lain 90 Cor	ntrol tasks 10	Training tasks 0			
Create from mai	n tasks			Download	
ID ¢	Overlap \$	Responses from performers \$	Correct responses, % \$	Last updated 💠	Distribution of correct responses for control tasks (
19f90ff6	00	0		07/26/2019 21:45:27	result
19f90ff0	00	0		07/26/2019 21:45:27	40% BAD
19f90ff1	00	0		07/26/2019 21:45:27	60% OK

• Save the markup and check the number of control tasks.

▶ test pool — clos	ed	Statistics 🛃 Download results 💙 Edit 🗸	· 0
POOL TASKS (File example for task uploading (t	sv, UTF-8)) 💿		
1 Upload B files Delete	Edit O Preview	0 %	
~30 task suites	0 training task	Done 0	
90 tasks	10 control task	0 ~	30

4. Start the pool.

Important. Remember that real Toloka performers will complete the tasks. Double check that everything is correct with configuration of your project before you start the pool.

test pool — clos	ed			Statistics 🕹 Download n	esults 🗸	Edit	~ 0
POOL TASKS (File example for task uploading (t	sv, UTF-8))	Preview		0 %			
~30 task suites	0 training task			Done 0			
90 tasks	10 control task		0				~30

Receiving responses

Disclaimer: Aggregation takes from 5 to 20 minutes. During this time, you can start configuring your next project. Refresh the Operations page to check progress.

- 1. Wait until the pool is completed. Refresh the pool page to check progress.
- 2. Click the arrow next to the **Download results** button and run aggregation using the **Dawid-Skene model**.

rojecto > boco the image contains trainen	gitts. 7 poor			M	
				Statistics Download results Edit View operations	~ 0
POOL TASKS (File example for task uploading	g (tsv, UTF-8)) 💿			Dawid-Skene aggregation model Aggregation by skill	
▲ Upload 🖺 files	Edit	• Preview	2	100 %	
30 task suites	0 training task			Done 30, accepted 30	
90 tasks	10 control task		0	View assignments	30
			-		

3. Go to the operations list and wait until aggregation finishes. **Note.** During this time, you can start working on your next project. Refresh the Operations page to check progress.

Yandex Toloka	Projects
Assignment aggregation started successfully. <u>View operations list</u> .	я фия ⊔тров → to
4. When aggregation is complete, download the TSV file with the results.	

PROJECT POOL test pool 10 v Id \$ Type \$ Started \$ Finished \$ Progress Start Files 0f1301 Dawid Skene aggregation model 06/26/2019 13:52:16 06/26/2019 13:57:12	Operatio	ons					
Id \$ Type \$ Started \$ Finished \$ Progress Started \$ 0f1301 Dawid-Skepe aggregation model 06/26/201013:52:16 06/26/201913:57:12 100% Success Download	PROJECT	POOL ✓ test pool	10 .				
0f1301 Dawid Skepe aggregation model 06/26/2010 12:52:16 06/26/2010 12:57:12 100% Success Download	Id ¢	Type \$	Started \$	Finished \$	Progress	Stor	Files
	0f1391	Dawid-Skene aggregation model	06/26/2019 13:52:16	06/26/2019 13:57:12	100%	Success	Download

5. Use this file to prepare data for Project #2.

Project #2 Find a similar item in an online store



Important: If you just started using Toloka, make sure that the English language is selected.

Yandex Toloka	Projects Users Skills	Profile Messages	② \$0.00 / \$5.59 ☵ Toloka requester
Projects Active ~	Only mine v		
Title 💠	Private comment \$	Created \$	Open pools
		No data	

Project creation

Disclaimer: In this particular example we ask the performers to search for the clothing items on <u>Marks and Spencer</u>. However, you can use any other online store. In this case you must change Marks and Spencer to the store of your choice in Specifications, HTML and JS fields of the project.

1. Click the **+ Create project** button and choose the **Blank** template.

Use this option to create a project from scratch if your task
doesn't materiary of the templates above.
Select

2. Enter a clear project name and description. It will be shown to performers.

PROJECT NAME	Find same or similar shoes on M&S	<
DESCRIPTION	Go to the M&S online and find the same or similar shoes on the website \qquad	<
INSTRUCTIONS 🕜		>
	Look at the shoes a person in the picture is holding or wearing in the picture. Go to the M&S online and find the same or similar shoes on the website. The shoes should be similar in color, style and height.	5
	Example:	

- 3. Write short and simple instructions.
- 4. Define specifications for the input and output data:

Input:

• The "image" input data field with the link type will be used to pass the image links to the performers.

Output:

- In the output field add "found_image", to give the performers space to upload the image.
- In the output field add "**button**" to check if the performer actually went to the required website
- In the output field add "found_link", here we will check whether the pattern of the url from the required website (a.g. online store, Marks and Spencer) matches the pattern of the submitted link. If you have chosen a different store then change "pattern": "https://www.marksandspencer.com/.*", to "pattern": "your_store.*"

The code for specifications are:

Input data:

{

{

```
"image": {
    "type": "url",
    "hidden": false,
    "required": true
}
```

Output data:

```
"button": {
  "type": "boolean",
  "hidden": false,
  "required": true,
  "allowed values": [
    true
 ]
},
"found link": {
  "type": "string",
  "hidden": false,
  "pattern": "https://www.marksandspencer.com/.*",
  "required": true
},
"found_image": {
  "type": "file",
  "hidden": false,
  "required": true
}
```

- 5. Create the task interface.
- Delete the whole HTML code in the template, and instead add the following code to show the initial item to the performers:

```
{{img src=image width="50%" height="400px"}}
<div class='answers'>
   Find the same <b>shoes</b> on Marks and Spencer
   {{field type="button-clicked" name="button" label="Marks and Spencer"
href="https://www.marksandspencer.com" action=true}}
   Shoes must be the same color and the same style.
   Paste the link here
   {{field width="100%" type="input" name="found_link"}}
   Upload the image here
   <div>
    {{field width="100%" type="file-img" name="found_image" preview=true}}
   </div>
  </div>
```

6. Now we need to check whether performer is going to submit a valid link and an image. To check it, <u>DO NOT delete any of JS code</u>

In case you are having trouble previewing call our team or check the expanded code in Appendix.

7. Add the following code in CSS field to set the images sizes proportionally:

```
.task {
   display: block;
   height: 500px;
   width: 800px;
}
.img {
   float: left;
   width: 50%;
}
.answers {
   float: left;
   width: 40%;
   margin: 5%;
}
```

8. Click the **Preview** button to see the performer's view of the task. In this particular case you will not be able to submit the assignment in **Preview** as the image you will be trying to upload cannot be uploaded while the pool is still closed.

|--|

9. Click Save button to save the project.

Pool creation

1. Click Add pool.

Select traf	fic lights b	y a bounding	box — active				Project actions $ \smallsetminus $
Statistics for 7 da	Spent	Ouality: main tasks	Ouality: training tasks	Average submit time	Users	Banned users	
0	0\$			-	0	0	
Pools Training	g Statistics	Quality control			_		
	Sea	Filters					Add a pool
Title 💠	Priority \$	Pr	ogress	Status 🔶		Started ¢	To be completed
		A	dd a pool to upload task	s, filter users, and set o	quality control rul	es.	

- 2. Give the pool any convenient name and description. You are the only one who can see them.
- 3. Specify the pool parameters:
 - Price per task page (for example, \$0.01)

	Price per task	suite		
	You can add one or i page.	more task	ks to the page. Enter the to	tal price for all tasks on the
PRICE IN US DOLLARS 🕜	0.01	×	FEE 🕜	0.005
	+ Dynamic pricing]	-	

- 4. Set up user filters.
 - Select English-speaking performers using the Language = English filter.

lser filter 💿				Copy settings from
PROFILE		_	_	
Languages	~	=	English	× 🗎 +
			-	

• Create the **"Found_shoes"** <u>skill</u> that will be assigned to users after they complete the pool tasks. You will use this skill to prevent these users from checking tasks in the next project. Click **Create skill:**

Users filter 💿	Copy settings from
Add filter	✓ Create skill

• Enter the skill name and add a description if desired. You are the only one who will see it. Leave the skill **private**, as it is by default and click **Add**.

Add skill	
TITLE	
Found_shoes	\times
DESCRIPTION	
Public? No	
	Cancel Add

5. Turn on the **Non-automatic acceptance** option and enter the number of days for checking in the **Deadline** field (for example, 7).

		Quality control			
		Add rules to get more All rules work indepe	e accurate responses. Indently.		
	NON-AUTOMATIC ACCEPTANCE	Yes	REVIEW PERIOD IN DAYS	7	\times
	CAPTCHA FREQUENCY 📀	None	~		

- 6. Set up <u>quality control</u>:
 - Resend the rejected tasks for completion. Add the <u>Recompletion of rejected</u> assignments:

RECOMPLET	ION OF REJECTED ASSIGNMENTS 🕖	
lf	assignment becomes v rejected v +	ŧ.
then	extend overlap by V 1 X	
	Open pool if closed	

• Submitted responses. Add a rule to mark users who completed at least one task in the pool.

SUBMITTED	RESPONSES 🕖			
lf	submitted assignments	~	2 1 × +	
then	assign skill value	~	found_shoes \times 1 \times	+

Tip. If the skill you created doesn't appear in the drop-down list, save the pool, and then open it for editing again.

Add Fast responses rule to block those who provide information suspiciously fast

Recent values to use items Minimum time per task suite 60 × If number of fast responses v ≥ 1 × + then ban v on project v 10 × days v fast responses	FAST RESPONSES 💿	
If number of fast responses then ban 10 × days fast responses ↓	Recent values to use items Minimum time per task suite 60 ×	Ť
10 × days v	If number of fast responses v 2 1 × + then ban v on project v	
+	10 × days × fast responses	× +

 Add <u>Results of assignments review</u> rule to ban those who brought results of improper quality

RESULTS OF	ASSIGNMENT REVIEW 💿	
Recent	values to use items	
lf	rejected responses (%) v 2 1 × +	
then	ban v on requester v	
	20 × days v	
	rejected assignment	
		+

Optionally, add other quality control rules.

Tip. Control tasks and majority vote are not used in this type of project, because performer's links and photos that she will provide must exactly match the reference, which is practically impossible.

7. Overlap. This is the number of users who will complete the same task. Because we want various options for each photo, put overlap equal to 3.

	ĵ	Overlap
		Specify how many performers you want to complete each task in the pool.
	OVERLAP 🕜	3 ×
DYNA	AMIC OVERLAP 🕐	Off

8. Optionally, specify the percentage of top-rated performers in the Speed / Quality ratio.

Speed/quality ratio 💿								
Top %	TOP N	Time						
Specify the	e percenta <u>c</u>	e of top-rat	ed active u	users who	an access	s tasks in	the pool.	
3523 ∱ Speed	All 9	1 ' I ' 0% 80%	70%	60% 509	40%	30%	1 1 1 20% 10%	352 🗼 Quality
60% top-rated performers were selected. The task is available to 2113 active users.								

Important: This can slow down pool completion.

9. Time allowed for completing a task page (for example, 300 seconds)

I	Parameters		_		
TIME FOR COMPLETING A 7	300	\times	POOL CLOSING DATE 💿	2021-06-04	
KEEP TASK ORDER 📀	No	TIME	BEFORE POOL CLOSES OIN SECONDS	0	
		POO	L PRIORITY IN PROJECT 💡	0	

10. Save the pool.

Preparing and uploading a file with tasks

- 1. Open the file with aggregated results from the project #1.
- 2. Select only images suitable for highlighting (**OK** answers or another value if you have changed it in the **"result"** field). Use a text editor or a spreadsheet editor.
- 3. Copy the column with the selected links to a new page or document and give a name to the **INPUT:image** column.

Important: If you changed the input field name in the project to something other than "image", change the name in the file as well: INPUT:<your field name>.

- 4. Save the file in TSV format.
- 5. <u>Upload the file</u> to the pool by selecting **Set manually**. Set 1 task per page.

Settings for file upload							
Tasks per page							
By empty row	Set manually		Smart	L L L L L L L L L L L L L L L L L L L			
Tasks per page		1]		×			
File example for task uploa		Close	Upload				

6. Start the pool.

Project #3 Does the item found look similar to the initial one?



Important: If you just started using Toloka, make sure that the **English** language is selected.

Yandex Toloka	Projects Users Skills	Profile Messages	
Projects Active ~	Only mine $$		Find project + Create project
Title 💠	Private comment \$	Created 🔶	Open pools
		No data	

Project creation (similar to the 1st project)

1. Click the button + Create project

Yandex Toloka	Projects Users Skills Profile Messages	🕜 \$0.00 / \$5.59 🚟	Toloka requester
Projects Active ~	Yandex ~	Find project	+ Create project
Title 🗧	Private comment 💠	Created 🗘	Open pools

2. Choose the **Image classification** template.

Templates							
Use templates to create and publish tasks that meet your needs. You can use a template as-is or adjust it to your input data and response format.							
Classification							
	Video moderation Users watch a video and choose one of the options. The task has a video player and several radio buttons.		Image classification Good for image classification and tagging. The template includes an image and several radio buttons.				
©	Select O Preview		Select O Preview				

3. Enter a clear project name and description. Important: It will be visible for real people.

NAME	Do the shoes look similar?	\times
ION	Take a look at two pairs of shoes and decide whether they look similar or not.	×
INSTRUCTIONS ⁽²⁾	fT ∰ B I U ■ GD ≔ ≔ Ξ Ξ Ξ Ξ Ξ Ξ []	• • •
		-
	Take a look at the pictures, which will show two pairs of shoes. Decide whether look similar or not.	they
	Take a look at the pictures, which will show two pairs of shoes. Decide whether look similar or not. The shoes will look similar if they are the same or similar color, fabric, length and	they style.

- 4. Write short and simple instructions.
- 5. Define parameters for the input and output data:

Input:

- The "image" input data field with the "url" type will be used to pass the initial image links to the performers.
- The **"found_link"** field with the "url" type will allow performers can go to the website.
- The **"assignment_id"** field with the "string" type, will be used to pass the number of the completed task.

Output:

• Leave "result" as it is.

The code for specifications is:

Input data:

{

}

```
"image": {
   "type": "url",
   "hidden": false,
   "required": true
},
   "found_link": {
    "type": "url",
    "hidden": false,
    "required": true
},
"assignment_id": {
    "type": "string",
    "hidden": true,
    "required": true
}
```

Output data:

```
"result": {
    "type": "string",
    "hidden": false,
    "required": true
}
```

6. Delete the whole HTML code in the template, and add the following one.

```
{{img src=image height="400px"}}
{{iframe src=found_link height="600px"}}
    Check that the uploaded image matches the product in the store.
      {{button label="Check the item" href=found_link action=true}}
    Are these <b>shoes</b> similar to each other?
      Shoes must be the same color and the same style.
      {{field type="radio" name="result" value="Yes" label="Yes"}}
      {{field type="radio" name="result" value="No" label="No"}}
```

</div>

- 7. Leave the JS block unchanged.
- 8. In the CSS block paste the following (don't forget the other half on the next page):

```
.task {
display: block;
min-height: 620px;
width: 100%;
box-sizing: border-box;
width: calc(100% - 100px);
}
.img {
float: left;
width: 30%;
}
```

```
.iframe {
 float: left;
width: 48%;
margin-left: 10px;
}
.text {
 float: left;
width: 18%;
margin-left: 10px;
}
```

9. Click the **Preview** button to see the performer's view of the task.

WINTEDIA	1.00 Anno 100 Nexed Voue-score Voe-100 O	nggelin Tak Gerebulans Deals assort Lag h ay [Search Wikipedia Q.]	Check that the uploaded image matches the product in the store.
	Territory in the interpretent of the interp	An and be represent affinituation	Check the item Are these shoes similar to each other? Shoes must be the same color and the same style. Yes No

- 10. Select the radio buttons in the preview and make sure that the task can be completed.
- 11. Click Save button to save the project.
 Note. To edit project parameters, click the button in the list of projects or Project actions → Edit on the project page.

Pool creation

- 1. Click Add pool.
- 2. Give the pool any convenient name and description. You are the only one who can see them.
- 3. Specify the pool parameters:
 - Set the price per task page (for example, \$0.01).

	Price per task suite		
	You can add one or more tasks	to the page. Enter the tot	al price for all tasks on the
PRICE IN US DOLLARS 😨	0.01	FEE 📀	0.005
	+ Dynamic pricing		

- 4. Set up user filters.
 - Select English-speaking performers using the "Language = English" filter. Prevent performers who completed previous tasks from checking this one. To do this, set a filter with the <u>"Found_shoes" skill</u>:

The "Found_shoes" skill = absent (empty field)

filter 🥥		Copy settings fr
Add filter	~ Create :	skill
PROFILE	✓ = English	× = +
PROFILE Languages SKILLS		X = +

Optionally, specify the percentage of top-rated performers in the <u>Speed / Quality ratio</u>. Important: This can slow down pool completion.

Speed/qu	ality ratio 💿						
Top %	Online	Time					
Specify the	percentage of	top-rated ac	tive users w	ho can access	tasks in the po	ol.	
3523 ∱ Speed	All 90%	1 1 1 80% 709	60%	1 1 1 1 50% 40%	30% 20%	' I 10%	352 ∱ Quality
60% top-ration The task is a	ed performers available to 211	were selecte 3 active use	d. 's.				

 Set up <u>quality control</u>: <u>Golden Set aka Control tasks</u>. Ban performers who give incorrect responses to control tasks. Example:

Recent	values to use items			i
lf	number of responses	~	≥ 3 × +	
and	correct responses (%)	~	< 60 ×	
then	ban	~	on project 🗸 🗸	
	10 ×			
	Reason (visible to you onl	V)		

Fast responses. You can ban the performers who suspiciously fast responses. This way you can get rid of cheaters in your pool. Example:

TAST REST C		
Recen	values to use litems	
Minim	um time per task suite 60 ×	
lf	number of fast responses v > 1 × +	
then	ban v on project v	
	10 ×	

6. Overlap. This is the number of users who will complete the same task. For example, 3 is enough for aggregation.

ecify how many performers	s you want to complete each task in the pool.
3 ×	
Off	
	B Coff

- 7. Time allowed for completing a task page (for example, 300 seconds).
- 8. Keep task order for your convenience

		Parameters	
2021-06-04	POOL CLOSING DATE 🕜	300 ×	TIME FOR COMPLETING A 7 TASK PAGE IN SECONDS.
0	ME BEFORE POOL CLOSES IN SECONDS	Yes	KEEP TASK ORDER 📀
0	OOL PRIORITY IN PROJECT 🔞		

9. Save the pool.

Preparing and uploading a file with tasks

- 1. Wait until the pool of project #2 on "finding similar shoes" is completed.
- 2. Open the pool page in Project #2 and click the **Download results** button



- Clear the Accepted checkbox and select Submitted.
- Clear link, user ID, status, start time and Separate assignments with empty row checkboxes. This will give you a list of unreviewed tasks.

Download results							
Status	Active	Submitted	Accepted				
	Rejected	Skipped	Expired				
Columns	🗌 link	dassignment id	🗌 task suite ID				
	🗌 user ID	status	start time				
	submit time	reject time					
	skip time expire time price						
Download	Download data for the period						
Separate a	Separate assignments with empty row						
Exclude assignments by banned users							
Close Download results							

- 3. Keep and rename the following columns:
- Keep the name of the "INPUT:image" column as it is.

- Change the name of the "OUTPUT:found_link" column to "INPUT:found_link". To check this image for correctness in project 3.
- Change the "ASSIGNMENT:assignment_id" column name to INPUT:assignment_id" to later track and match the assignment number.

Make sure	vou havo th	no hoadars	of the columns	exactly as below
Make Sule	you nave u	ie lieauei s	or the columns	Exactly as Delow

А	В	С
INPUT:image	INPUT:found_link	INPUT:assignment_id
https://tlk.s3.ya	https://www.asos.c	https://www.asos.com/a
https://tlk.s3.ya	https://www.asos.	https://www.asos.com/r
https://tlk.s3.ya	https://www.asos.	https://www.asos.com/r
https://tlk.s3.ya	https://www.asos.c	https://www.asos.com/r

Save the file in TSV format.

- 4. Open the pool page in Project #3.
- 5. Upload the file to the pool by selecting **Smart mixing**. Set the number of main and control tasks per page (for example, 9 and 1).

Settings for file up	load 🔍		
Tasks per page			
By empty row	Set manually	s	imart mixing
Main tasks		9	×
Training tasks		0	
Control tasks		1	×
Show advanced setting	5		
File example for task up	loading	Clos	e Upload

6. Create control tasks.

Note. Control tasks are tasks with the correct response known in advance. They are used to track the performer's quality of responses. The performer's response is compared to the response you provided. If they match, it means the performer answered correctly.

Click Edit \rightarrow Create control tasks.

test pool — clo	osed		Statistics	▲ Download results	~	Edit	~	
POOL TASKS (File example for task uploadin	g (tsv. UTF-8))		0	%				
0 task suites	0 training task			Done 0				
100 tasks	O control task	0					0	

Edit tasks							
Use main tasks as a starting point to create control tasks or training tasks.							
Control tasks are for checking the quality of responses from performers. They contain correct responses to compare with actual responses.							
Training tasks are for	teaching performer	s how to complete tasks. They cont	ain correct responses and hints.				
Learn more							
	cability of the maining						
Create control tasks	Create training task	ks	Download				
Create control tasks	Create training task	ks Responses from performers \$	Download				
Create control tasks ID \$ 1975eb01	Create training task Overlap \$	ks Responses from performers \$ 0	Download				
Create control tasks ID \$ 1975eb01 1975eabe	Create training task Overlap \$ 3 3	ks Responses from performers ¢ 0 0	Download Last updated + 07/26/2019 22:18:06 07/26/2019 22:18:06				

• Check the "result" output field that is used to match the user response to the control one, select the response and click **Save and go to next**.

Projects \rightarrow Do the shoes in the picture look \rightarrow Do t	the shoes in the picture look $_{\rm ent} \rightarrow $ Uploaded tasks	- Edit tasks		Control tasks: 6	0	$\langle \rangle$
Projects → Do the shoes in the picture look → Do the picture look.	he shoes in the picture look) Uploaded tasks		Check that the uploaded image matches the product in the store. Check the tem tem the store of the same color and the same style. Yes • No	LOURDO LASKS 6 II ID 07c90746 Main 07c90747 Main 07c90743 Main 07c90743 Main 07c90743 Main 07c90743 Main 07c90743 Main 07c90740 Main	Details Overlap: 3 Responses: 3 Responses: 3 Overlap: 3 Responses: 3 R	< > Updated: 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020 06/04/2020
Bave and go to next				Main	Responses: 3	

Enter correct responses for your control tasks. In small pools, control tasks should account for approximately 10% of all tasks.

Tip. Make sure to include different variations of correct responses in equal amounts. Open the **Control tasks** \rightarrow **Distribution of correct responses for control tasks** tab.

Edit tasks							
Use main tasks as a starting point to create control tasks or training tasks.							
Control tasks are for compare with actual	checking the quality responses.	of responses from performers. The	y contain correct responses to				
Training tasks are for	teaching performer	s how to complete tasks. They cont	ain correct responses and hints.				
earn more	learn more						
Main 90 Control t	asks 10 Training Create training task	p tasks 0	Download				
Main 90 Control t Create control tasks	asks 10 Training Create training task Overlap ¢	tasks 0 ks Responses from performers \$	Download Last updated 🔸				
Main 90 Control t Create control tasks ID ¢ 19f9102c	asks 10 Training Create training task Overtap ¢ 3	tasks 0 ks Responses from performers \$ 0	Download Last updated 07/26/2019 21:45:27				

Main 90 Co	ntrol tasks 10	Training tasks 0			
Create from ma	in tasks			Download	
ID ¢	Overlap \$	Responses from performers \$	Correct responses, % 💠	Last updated 💠	Distribution of correct responses for control tasks 📀
19f90ff6	00	0		07/26/2019 21:45:27	result
19f90ff0	00	0		07/26/2019 21:45:27	40% BAD
19f90ff1	00	0		07/26/2019 21:45:27	60% OK

• Save the markup and check the number of control tasks.

▶ test pool — clos	ed	Statistics 🕹 Download results 🗸 Edit 🗸	0
POOL TASKS (File example for task uploading (t	sv, UTF-8)) 🔘		0
1 Upload h files 1 Delete	Edit • Preview	0 %	
~30 task suites	0 training task	Done 0	
90 tasks	10 control task	0 -3	0

7. Start the pool.

Important. Remember that real Toloka performers will complete the tasks. Double check that everything is correct with configuration of your project before you start the pool.

test pool — clos	sed	Statistics 🛓 Download results 💙	Edit 🗸 🔘
POOL TASKS (File example for task uploading (tsv. UTF-8)) 🔘		
1 Upload 1 files 1 Delete	Edit O Preview	0 %	
~30 task suites	0 training task	Done 0	
90 tasks	10 control task	0	~30

Receiving responses

Disclaimer: Aggregation takes from 5 to 20 minutes. During this time, you can start configuring your next project. Refresh the Operations page to check progress.

- 1. Wait until the pool is completed.
- 2. Click the arrow next to the **Download results** button and run aggregation using the **Dawid-Skene model**.



3. Go to the operations list and wait until aggregation finishes.

Yandex Toloka	Projects
Assignment aggregation started successfully. <u>View operations list</u> . Projects Singht Coop res	арлык фильтров — К

4. Download the responses.

Operatio	ons					
PROJECT	POOL v test pool 1	0 ~				
ld ¢	Туре 💠	Started \$	Finished \$	Progress	Stor	Files
0f1391	Dawid-Skene aggregation model	06/26/2019 13:52:16	06/26/2019 13:57:12	100%	Success	Download

Upload reviewed results

As you set **post verification** in the <u>pool settings in Project #2</u>, you need to check the performers' responses within the time limit set in the **Deadline** field.

- 1. Open the file with aggregated results in a spreadsheet editor.
- 2. Add the following columns:

- "ACCEPT:verdict" — The result of verification.

- "ACCEPT:comment" — Comments for performers if responses were rejected (for example, which part of the instructions wasn't followed).

- 3. Change the name of the "INPUT:assignment_id" column to "ASSIGNMENT:assignment_id".
- 4. Delete all other columns
- 5. Fill in the "ACCEPT:verdict" and "ACCEPT:comment" columns:
 - If the aggregated result for the task is OK, put "+" then the task will be accepted.
 - If the result is BAD or 404, put "-" then the task will be rejected. Enter the reason for rejection in the "ACCEPT:comment" field . For example: *The item provided is incorrect or improper.*
- 6. Now you can delete the other columns. Save the edited TSV file.

А	В	С
ASSIGNMENT:assignment_id	ACCEPT:verdict	ACCEPT:comment
00009c42455e27381d139baa	+	
00009c42455e27337bb0e86f	-	The item provided is incorrect
00009c42455e2737692f3cb80	+	
00009c42455e2739bcb0e86f0	+	

- 7. Open the pool page in Project #2.
- 8. Click <u>Review assignments</u> on the pool page above the progress bar.
- 9. Click Upload review results.

bmitted respo	onses				4	b Down	nload resul	lts 🗸	1 Uploa	d review results
30 All assignments		30 Under review		O Accepted assignments			0	0 Rejected assignments		0
Accept ⊘ Reject	Actions v						Submitted	Accepted	Rejected	All assignment
Accept 🕥 Reject Response 🔻	Actions 🤟	User ¥	Comp	leted 🕈	Duration		Submitted	Accepted	Rejected	All assignment
Accept Response	Actions	User ₹ 483927a3f25b15f6a85c7f97d2a1c9f	Com;	leted • /2020 21:08:57	Duration 1 min 5 sec		Submitted	Accepted	Rejected	All assignment Statu
Accept @ Reject Response ¥ 00009e68cb-5e2dd3 00009e68cb-5e2dd3	C Actions ↓	User T 483927a3f25b15f6a85c7f97d2a1c9f1 483927a3f25b15f6a85c7f97d2a1c9f1	Comp 5 01/26 5 01/26	Heted ♦ /2020 21:08:57 /2020 21:09:36	Duration 1 min 5 sec 38 sec		Submitted	Accepted	Rejected	All assignment Statu

- 10. Select the file and upload it to Toloka.
- 11. Check that all tasks have changed their status to accepted or rejected.

Assignments	5			▲ Download n	results
65		Assigments review (to_accept.tsv)			0
All assignments		REVIEW	RESULTS		Rejected assignmen
		63 accepted	2 rejected		
Accept 🥥 Re	iject 👩 Actions 🗸	0 status not changed	65 processed successfully	itted	Accepted

12. You rejected tasks and set up the rule to send them for re-completion while configuring Project 2.

If have enough time you can do as many more reiterations as needed in order to receive as much clean data as you can. The steps are the following: The pool will open again, and these tasks will be resent to other performers. After the pool is marked up, download the new results and submit them for review. Download the reviewed results. Repeat these steps until all the images from the second project are correctly marked up.

But if you do not have enough time, move on to the next project and you can complete the reiterations later at your own pace.

Review assignments in the interface (another way of results validation)

You can also <u>review assignments by yourself</u> and see the results of the crowdsourcing pipeline that you have created.

- 1. Open the pool page in Project #2.
- 2. Click the **Review assignments** button on the pool page.

Accept 🥏 Reject 😨 Actions 🗸				Submitted	Accepted	Rejected	Allassignments
Assignment y	User y	Date ≑	Durat.				Status
0000586e7e5d2c9c446064f701220ca9c5	788168bc6cebfbaade32a38035e505b9	07/15/2019 18:32:06	49 sec	-			
0000586e7e5d2c9c3ab5cff0011e313fa0	e85a361c96450663de00c64a12d9385f	07/15/2019 18:32:28	1 min 21 sec	_			•

- Choose an assignment then click Accept or Reject.
- For rejected assignments, enter a comment (explain why you decline it).

Project #4 Which item is more similar?



Important: Before you start using Tolok a, make sure that the **English** language is selected.

Yandex Toloka	Projects Users Skills	Profile Messages	② \$0.00 / \$5.59 ₩ Toloka requester
Projects Active V	Only mine $$		Find project + Create project
Title \$	Private comment \$	Created 🔶	Open pools
		No data	

Project creation

1. Click the button **+ Create project**

Yandex Toloka	Projects Users Skills Profile Messages	🕜 \$0.00 / \$5.59 🚟	Toloka requester
Projects Active ~	Yandex ∨	Find project	+ Create project
Title 🗧	Private comment \$	Created 🗘	Open pools

2. Choose the **Side-by-side image comparison** template.

Side-by-side image comparison For comparing two images. The template contains two images and several radio buttons.
Select O Preview

- 1. Enter a clear project name and description. Important: It will be visible for real Toloka performers.
- 2. Write short and simple instructions. Example:

PROJECT NAME	Which shoes look more similar?			\times
DESCRIPTION	Decide which pair of shoes look more alike to the initial one			\times
INSTRUCTIONS	τĨ Φ Β Ι U Β Θ Ξ Ξ Ξ Ξ	13	0	$\langle \rangle$
	What you need to do:			
	Look at 2 pictures with different shoes and decide which pair of shoes look mo to the initial pair. Use your own sense of style, but also remember that they wi alike if they are similar color, similar form, similar fabric and similar length :)	ore s Il loc	imi k	lar
	Good luck!			

- 11. Define parameters for the input and output data:
 - The "image" field is the initial image from the dataset.
 - The "left_link" field is a link that the performer provided to match the item from the initial image.
 - The "right_link" field is another link the performer provided.

You will be able to upload the file with links to the pool later.

• The "result" field will be used to receive performer's responses.

The code for specifications is: Input data:

```
{
  "image": {
    "type": "url",
    "hidden": false,
    "required": true
  },
  "left_link": {
    "type": "url",
    "hidden": false,
    "required": true
  },
  "right link": {
    "type": "url",
    "hidden": false,
    "required": true
 }
}
Output data:
{
  "result": {
    "type": "url",
```

```
"type": "url",
"hidden": false,
"required": true
}
```

12. Create the task interface in the HTML block.

```
<div class="header">
    <div class="left caption">
       {{button label="Go to site" href=uploaded link left size="L"}}
       {{uploaded link left}}
   </div>
    <div class="right caption">
       {{uploaded link right}}
       {{button label="Go to site" href=uploaded link right size="L"}}
   </div>
</div>
{{img src=image}}
<div class="content clearfix">
   <div class="left page">
      {{iframe src=uploaded_link_left width="100%" height="700px" real-
size=true screenshot=true}}
   </div>
   <div class="right page">
       {{iframe src=uploaded link right width="100%" height="700px" real-
size=true screenshot=true}}
   </div>
</div>
<div class="footer">
   {{field type="radio" name="result" label="The left one is better"
value=result_left hotkey="1"}}
   {{field type="radio" name="result" label="The right one is better"
value=result_right hotkey="2"}}
</div>
```

13. DO NOT delete any of JS code but add the following JS block right before OnRender

```
getTemplateData: function() {
       var data = TolokaHandlebarsTask.prototype.getTemplateData.apply(this,
arguments),
        input = this.getTask().input_values;
       var left link = input.left link;
       var right link = input.right link;
       var uploaded link left = '',
           uploaded link right = ''
        if (Math.floor(Math.random() * 2)) {
            uploaded link left = left link
            uploaded link right = right link
        } else {
            uploaded_link_left = right_link
            uploaded_link_right = left_link
        }
       data.uploaded_link_left = uploaded_link_left;
       data.uploaded_link_right = uploaded_link_right;
       data.result_left = uploaded_link left;
       data.result_right = uploaded_link_right;
        return data;
```

},

14. In the CSS block add:

```
.task {
 display: block;
  text-align:center;
}
.header {
 overflow: hidden;
 background-color: #FFCC00;
}
.caption {
 width: 50%;
1
.url {
 white-space: nowrap;
 overflow: hidden;
 text-overflow: ellipsis;
 max-width: calc(100% - 182px);
 display: inline-block;
 vertical-align: bottom;
}
.button {
margin: 10px;
 max-width: 182px;
}
.button_label {
 white-space: nowrap;
 overflow: hidden;
 text-overflow: ellipsis;
 max-width: 150px;
}
.content {
```

```
margin: 10px 0;
}
.page {
 display: inline-block;
 width: 50%;
}
.left {
 float: left;
 text-align: left;
}
.right {
float: right;
 text-align: right;
}
.clearfix {
 overflow: hidden;
 width: 100%;
}
```

15. Click the **Preview** button to see the performer's view of the task. You will see standard pictures on the page.

Go to site https://	/wikipedia.org/wiki/0					ht	ps://wikipedia.org/wiki/0	Go to sh
1.000		L Heriog	ges in Talk Contributoria Craate account Log in	1.000		L Not in	gost in Talk Contributions Create a	account Lo
1 0 × 1	Article Talk	Read View source . View history	Bearch Willipedia Q	10 .	Article Tak	Read View source View histor	y Gearch Wikipedia	
WIKIPEDIA	0		6	WIKIPEDIA	0			
an rate approximation and page and page	The theorem is a second of the encoder second of \$2.5 km of the to encoder second of \$2.5 km of the to the to encoder second of \$2.5 km of the to the to encoder second of \$2.5 km of the top	Bits active O or the East Active symbol Q → or Onexit replandinguistics. Intern III and IIII Active Act	and b specear all muture. 0 1 - 4	te ren das temperatures Granes de la constante Granes de la constante Constante de la constante Norte de la constante Reseaures Norte de la constante Reseaures Reseaures Norte de la constante Reseaures Re	The there encyclears: The observation of the encoder and off \$10 if is not to be only induced: there, for a status encoder and status, see No. 10 if is a status of the encoder and status, see No. 10 if is a status of the encoder and status of the encoder is induced: a status of the encoder and the encoder is the encoder and the encoder and the encoder is the encoder and the encoder and the encoder is the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder and the encoder is 10 if the encoder and the encoder is the encoder is 10 if the encoder and the encoder is the encoder is 10 if the encoder and the encoder is the encoder is 10 if the encoder and the encoder is the enc	and with the latter O or the East Asian symbol (), or O may of <i>Diror</i> (Jacobia) and <i>Diror</i> (Jacobia) and Asian and Asian and Asian Asian and Asian and Asian Asi	kunch be represent allmutation	1 nought.
Salp			The left core is bet	er 📴 🔘 The right one is b	eter			

16. Select the radio buttons in the preview and make sure that the task can be completed.17. Click **Save** button to save the project.

Note. To edit project parameters, click the button in the list of projects or **Project** actions \rightarrow Edit on the project page.

Pool creation

- 1. Click Add pool.
- 2. Give the pool any convenient name and description. You are the only one who can see them.
- 3. Specify the pool parameters:
 - Price per task page (for example, \$0.01)

	Price per task s	uite		
	You can add one or n page.	nore task	is to the page. Enter the tot	tal price for all tasks on the
PRICE IN US DOLLARS 📀	0.01	×	FEE 📀	0.005
	+ Dynamic pricing			

- 4. Set up user filters.
 - Select English-speaking performers using the Language = English filter.

User filter 💿		Copy settings from
PROFILE		
Languages	∽ = English	× 🗎 +

5. Set up <u>quality control</u>:

Fast responses. You can ban the performers who suspiciously fast responses. This way you can get rid of cheaters in your pool. Example:

Recent	values to use items			
Minimu	m time per task suite 30	~		
Minimu	m time per task suite 30	×		
lf	number of fast response	s 🗸 📏	3 × +	
lf	number of fast response	s 🗸 > :	3 × +	
lf then	number of fast response	s v > :	3 × +	
lf then	number of fast response	s v > :	3 × +	
lf then	number of fast response ban	s v > :	3 × +	
lf then	number of fast response ban days	s v > : v on p	3 × +	

Optionally, add other quality control rules.

6. Overlap. This is the number of users who will complete the same task. Put a larger number in this task. For example, 10.



7. Optionally, specify the percentage of top-rated performers in the <u>Speed / Quality ratio</u>. Important: This can slow down pool completion.

Speed/qua	ality ratio 💿					
Top %	Online	Time				
Specify the p	percentage of	top-rated ac	ive users who can	access ta	asks in the pool.	
3523 🔶 Speed	I I I I All 90%	- I - I 80% 709	60% 50%	40%	1 1 1 1 30% 20% 10%	9 352 ∱ Quality
60% top-rate The task is a	ed performers vailable to 211	were selecte 3 active use	ł. 5.			

8. Time allowed for completing a task page (for example, 300 seconds).

		rameters	Paran	
2021-06-04	POOL CLOSING DATE 📀	00 ×	300	TIME FOR COMPLETING A 7 COMPLETING A
0	TIME BEFORE POOL CLOSES IN SECONDS	25	Yes	KEEP TASK ORDER 📀
0	POOL PRIORITY IN PROJECT 📀			

9. Save the pool.

Preparing and uploading a file with tasks

- 1. Take the downloaded TSV file with validated responses from Project 3.
- Now you need to generate pairs for each INPUT:image so that you will be able to compare two found images with the initial one and decide which one is more similar than another.

You can either generate the pairs by hand, using MS Excel or any other editor or you can automatically do it. We recommend using Python and Jupyter Lab.

You can consult with our results https://tlk.s3.yandex.net/wsdm2020/SbS_Toloka_prep&aggr_data.ipynb

3. <u>Upload pool tasks</u> from this file.

Important: If you changed the name of the input field, change it in the file as well

Projects \rightarrow Which shoes look more similar?	> Which shoes look more similar?	
Which shoes	ook more similar? — close	Catistics Download results V Edit V
POOL TASKS (File example for task uploadin	g (tsv, UTF-8)) 🔘	
1 Upload		0 %
0 task suites	0 training task	Done 0
0 tasks	0 control task	0 0

<u>Upload the file</u> to the pool by selecting **Set manually** and specify the number of tasks per page.

You can experiment with the number of tasks!

4. Start the pool.

Important. Remember that real Toloka performers will complete the tasks. Double check that everything is correct with configuration of your project before you start the pool.

Receiving responses

- 1. Wait until the pool is completed. Refresh the pool page to check progress.
- 2. Download **the accepted** results. Select the URLs, user IDs and assignment IDs like in the picture below. Do not forget to clear "Separate assignments with empty row" box!

Status	Active	Submitted	Accepted
	Rejected	Skipped	Expired
Columns	VRL	🧹 assignment ID	task suite ID
	🗹 user ID	status	start time
	submit time	accept time	reject time
	skip time	expire time	price
Download d	lata for the period		
Separate as	signments with empty row		
Exclude ass	ignments by banned users		

3. Try to run Bradley Terry model on these results (you can consult with our results https://tlk.s3.yandex.net/wsdm2020/SbS_Toloka_prep&aggr_data.ipynb)

GOOD LUCK!!!

Appendix: Expanded code of the projects

Project 1

Specifications:

Input: Output: { "image": { "type": "url", "hidden": false, "required": true } }

HTML:

```
{{img src=image width="100%" height="400px"}}
<div>Are there <b>shoes</b> in the picture?</div>
<div> {{field type="radio" name="result" value="Yes" label="Yes"
hotkey="1"}} {{field type="radio" name="result" value="No" label="No"
hotkey="2"}}</div>
```

JS:

```
exports.Task = extend(TolokaHandlebarsTask, function (options) {
 TolokaHandlebarsTask.call(this, options);
}, {
 onRender: function() {
  // DOM element for task is formed (available via #getDOMElement())
 },
 onDestroy: function() {
  // Task is completed. Global resources can be released (if used)
 }
});
function extend(ParentClass, constructorFunction, prototypeHash) {
 constructorFunction = constructorFunction || function () {};
 prototypeHash = prototypeHash || {};
 if (ParentClass) {
   constructorFunction.prototype = Object.create(ParentClass.prototype);
 for (var i in prototypeHash) {
```

```
constructorFunction.prototype[i] = prototypeHash[i];
}
return constructorFunction;
}
```

Project 2

Specifications:

Input:

```
{
  "image": {
    "type": "url",
    "hidden": false,
    "required": true
  }
}
```

Output:

```
"button": {
   "type": "boolean",
   "hidden": false,
    "required": true,
    "allowed values": [
    true
   1
  },
 "found link": {
   "type": "string",
    "hidden": false,
    "pattern":
"https://www.marksandspencer.com/.*",
    "required": true
 },
 "found image": {
   "type": "file",
   "hidden": false,
   "required": true
 }
}
```

HTML:

```
exports.Task = extend(TolokaHandlebarsTask, function (options) {
  TolokaHandlebarsTask.call(this, options);
}, {
  onRender: function() {
    // DOM element for task is formed (available via #getDOMElement())
  },
  onDestroy: function() {
    // Task is completed. Global resources can be released (if used)
  }
});
function extend(ParentClass, constructorFunction, prototypeHash) {
    constructorFunction = constructorFunction || function () {};
    prototypeHash = prototypeHash || {};
    if (ParentClass) {
```

```
constructorFunction.prototype = Object.create(ParentClass.prototype);
}
for (var i in prototypeHash) {
   constructorFunction.prototype[i] = prototypeHash[i];
}
return constructorFunction;
```

CSS:

}

```
.task {
   display: block;
   height: 500px;
   width: 800px;
}
.img {
   float: left;
   width: 50%;
}
.answers {
   float: left;
   width: 40%;
   margin: 5%;
}
```

Project 3

Specifications:

```
Input:
```

```
{
    "image": {
        "type": "url",
        "hidden": false,
        "required": true
    },
    "found_link": {
        "type": "url",
        "hidden": false,
        "required": true
    },
```

```
Output:
```

```
{
   "result": {
    "type": "string",
    "hidden": false,
    "required": true
   }
}
```

```
"assignment_id": {
    "type": "string",
    "hidden": true,
    "required": true
}
```

HTML:

```
{{img src=image height="400px"}} {{iframe src= found_link height="600px"}}
Check that the uploaded image matches the product in the store.
{{button label="Check the item" href=found_link action=true}}
Are these <b>shoes</b> similar to each other?
Shoes must be the same color and the same style.
{{field type="radio" name="result" value="Yes" label="Yes"}}
{{field type="radio" name="result" value="No" label="No"}}
```

JS:

```
exports.Task = extend(TolokaHandlebarsTask, function (options) {
  TolokaHandlebarsTask.call(this, options);
}, {
  onRender: function() {
    // DOM element for task is formed (available via #getDOMElement())
  },
  onDestroy: function() {
    // Task is completed. Global resources can be released (if used)
  }
});
```

```
function extend(ParentClass, constructorFunction, prototypeHash) {
  constructorFunction = constructorFunction || function () {};
  prototypeHash = prototypeHash || {};
  if (ParentClass) {
    constructorFunction.prototype = Object.create(ParentClass.prototype);
  }
  for (var i in prototypeHash) {
    constructorFunction.prototype[i] = prototypeHash[i];
  }
  return constructorFunction;
}
```

CSS:

```
.task {
  display: block;
  min-height: 620px;
  width: 100%;
  box-sizing: border-box;
  width: calc(100% - 100px);
```

```
}
```

```
.img {
float: left;
width: 30%;
}
.iframe {
float: left;
width: 48%;
margin-left: 10px;
}
.text {
float: left;
width: 18%;
margin-left: 10px;
}
```

Project 4:

Specifications:

```
Input:
```

```
"result": {
{
                                         "type": "url",
 "image": {
   "type": "url",
                                          "hidden": false,
                                         "required": true
   "hidden": false,
                                      }
   "required": true
                                      }
 },
 "left link": {
   "type": "url",
   "hidden": false,
   "required": true
 },
 "right link": {
  "type": "url",
   "hidden": false,
   "required": true
}
}
```

HTML:

Output:

{

JS:

```
exports.Task = extend(TolokaHandlebarsTask, function(options) {
   TolokaHandlebarsTask.call(this, options);
}, {
   getTemplateData: function() {
       var data =
TolokaHandlebarsTask.prototype.getTemplateData.apply(this, arguments),
           input = this.getTask().input values;
       var left link = input.left link;
       var right link = input.right link;
       var uploaded link left = '',
           uploaded link right = ''
        if (Math.floor(Math.random() * 2)) {
            uploaded_link_left = left_link
            uploaded link right = right link
        } else {
           uploaded link left = right link
           uploaded link right = left link
        }
        data.uploaded link left = uploaded link left;
        data.uploaded link right = uploaded link right;
        data.result left = uploaded_link_left;
       data.result right = uploaded link right;
 return data;
},
   onRender: function() {
    // DOM element for task is formed (available via #getDOMElement())
   },
   onDestroy: function() {
    // Task is completed. Global resources can be released (if used)
   }
});
function extend(ParentClass, constructorFunction, prototypeHash) {
   constructorFunction = constructorFunction || function() {};
   prototypeHash = prototypeHash || {};
   if (ParentClass) {
       constructorFunction.prototype =
Object.create(ParentClass.prototype);
for (var i in prototypeHash) {
```

```
constructorFunction.prototype[i] = prototypeHash[i];
}
return constructorFunction;
}
```

CSS:

```
.task {
display: block;
text-align:center;
}
.header {
 overflow: hidden;
background-color: #FFCC00;
}
.caption {
width: 50%;
}
.url {
white-space: nowrap;
overflow: hidden;
text-overflow: ellipsis;
max-width: calc(100% - 182px);
display: inline-block;
vertical-align: bottom;
}
.button {
margin: 10px;
max-width: 182px;
}
.content {
margin: 10px 0;
}
.page {
display: inline-block;
 width: 50%;
}
.left {
float: left;
text-align: left;
}
.right {
float: right;
text-align: right;
}
.clearfix {
overflow: hidden;
```

```
width: 100%;
}
.image {
  display: inline-block;
  width: 50%;
}
```