Deliverable number: D3.1
Deliverable title: A set of crowdsourcing interfaces

Deliverable description: A set of crowdsourcing interfaces (M12). This deliverable will consist of a set of Human Intelligence Task design experimentally validated for object recognition in images, Validation of named entity extraction, image labeling. This will be available for tasks in WP5

Due date of deliverable: 31.12.17
Actual date of deliverable: Resubmitted 21.5.18
Authors: Alessandro Checco, Rehab Qarout
Project partners: University of Sheffield
Workpackage: WP3
Workpackage leader: Paul Clough
Dissemination Level: Public

Change Log

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Status</th>
<th>Author (Partner)</th>
<th>Description/Approval Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31.12.17</td>
<td>Final</td>
<td>University of Sheffield</td>
<td>Public</td>
</tr>
</tbody>
</table>
# Table of Contents

1 Introduction 3

2 Crowdsourcing Process 3

3 Related Work 6
   Psychological Factors 7
   Type of Tasks 8
   Task Graphical User Interface (GUI) 9
   Training Questions 10
   Length of the Microtask 11
   Ordering Effects 12

4 The Role of Microtask Work Environments 13

5 Crowdsourcing Interfaces in Fashion 15
   Task Description 15
   Task Instructions 15
   Task Graphical User Interface (GUI) 16
      Strategies on GUI design 19
   Training 19

5 Conclusion 20

Bibliography 21
Summary. This document will provide a report on a set of Human Intelligence Task (HIT) designs, experimentally validated for object recognition in images, validation of named entity extraction, and image labeling, with particular focus on entity linking for images in fashion. This is the result of FashionBrain T3.1 “Tailored Crowdsourcing Tasks”, based on the investigation of the effect of task features (e.g., aesthetic, complexity, etc.) on the quality of the results collected from the crowd, and will be a building block for WP5 “Fashion Analysis in Social Media Streams”.
1 Introduction

A crowdsourcing task is generally an automated task generated by a computer and published to a crowdsourcing platform via its Application Programming Interface (API). The crowd workers will perform the tasks at hand through a web interface and submit their result. An accurate design of crowdsourcing tasks allows for better quality and less expensive data collection. In this task we will investigate the effect of task features on the quality of the results collected from the crowd. We will focus in particular on creating user interfaces for the following tasks within the FashionBrain project: object recognition in images, validation of named entity extraction, and image labeling. We will show how the quality of a HIT is affected by the following aspects: task instructions, training, interface clarity, and overall task design. This is particularly important in situations where tasks are paid as workers will typically aim to minimise the time they spend on understanding what they are required to do.

The rest of the document is structured as follows: in Section 2, we describe in detail the crowdsourcing process. In Section 3, we present an extensive literature research, that will guide our task design approach. In Section 4, we present our analysis of work environments and the tools we developed to assist task design. In Section 5, we present our proposed solution for entity linking and images in fashion.

2 Crowdsourcing Process

Despite the fact that different types of tasks exist on crowdsourcing platforms, the process of implementing each task consists of basically the same stages. The mechanism of crowdsourcing works according to the following steps as shown in Figure 1: (1) Define the problem, (2) Collect data, (3) Design the task, (4) Launch the task online via a crowdsourcing platform, (5) Analyse the result, and, if the job accepted, (6) Send rewards to the workers.

![Fig. 1. Mechanism of crowdsourcing interaction with business companies.](image-url)
The process of crowdsourcing could be analysed from three different perspectives: the worker, the requester and the task.

The worker registers on a platform and performs some unpaid tasks in order to become qualified in certain skills that might be required. On the platform, the workers will find a list of jobs available along with the specified reward for completing it accurately. The online crowd is invited to an open call for everyone who is interested in providing solutions or performing the tasks on behalf of the company, which will name a price for each task. In a particular situation, the crowd could be limited by the imposition of some constraints, such as needing certain experience in a given area [Brabham 2008].

Fig. 2. Crowdsourcing process (UML activity diagram) from the worker perspective.

A number of recommendation systems appear to favour some workers for a specific task based on criteria, such as workers' history and their overall performance [Geiger and Schader 2014; Schnitzer et al. 2015; Yuen et al. 2015]. The worker will choose one of the listed tasks and attempt to complete it. They could decide at any point to leave the task or submit an answer if they succeed in completing it. The last stage of the worker process is receiving a response for the submitted job, either rejection or the pre-agreed reward (Figure 2).

The requester represents the company who will identify tasks or problems that need to be solved. The requester will gather the data and define the requirements, constraints, and output of the job and, for a long or complex task, the requester has to divide these task into smaller tasks (microtasks) which are released to the crowd online via one of the crowdsourcing platforms. For each task, the requester will determine a specific amount of time for the user to complete this task and submit it to the requester. When this time is up the requesters will analyse the quality of the received work and decide if the problem has been solved by the completed work or whether it should be rejected; based on this decision the worker will receive a response. This mechanism could vary from one requester to another (Figure 3).
The task goes through three stages as shown in Figure 4. The first is the design process "off-line", where the task will be outlined using one of the predefined templates provided by the platform or designed from scratch [Luz 2015]. In this stage, the data or the input will be fed in, and the parameters of the task will be set. The measures for quality control will be implemented at this stage to guarantee efficient result and detect spam or malicious workers. Moreover, the long/complex task will be decomposed in simple/micro tasks. For example, identify the face of a specific person from a picture of a crowd in a football stadium will be a long task to perform by one worker. Such a task is able to be divided into micro-tasks by cropping the picture to small pieces and crowdsourcing each piece as a simple independent task to workers.

To correctly define task complexity we need to consider, as explained in [Finnerty 2013] and [Sweller 1988] task structure, task interdependence, task commitment, and cognitive load.

The second stage ("on-line" in Figure 4) is execution, where the microtasks appear online and become available to workers. The implementation of this process could be in parallel when a microtask does not depend on the result of another one. Alternatively, the microtasks could be implemented sequentially one after another, where the result of one task becomes the input for the next task. In this on-line stage, the task could be paused if it requires any modifications and then continues running again. In some situations, the task could be done by a number of different users, and each could be paid if they complete the task successfully. In other cases, such as logo design, the task could be completed in different ways depending on workers’ understanding and creativity, and the payment would be given for the ‘best’ solution, as decided by the requester [Whitla 2009].

The last stage is when the requester receives the completed job and it reaches either a "finished" or "cancelled" state. In the case of reaching a finished state, the microtasks will go through the aggregation process where these small tasks will be merged together to form the final result of the job. A post-execution quality control methods will be used to identify the cheater workers based on their performance on the submitted job and if they meet the quality criteria that have been set up in the pre-execution quality control method.
3 Related Work

The aim of this section is to present a comprehensive survey on crowdsourcing task design, focusing on the technical factors that have a significant impact on the quality of the design. This is a novel endeavour, because even the most recent surveys on crowdsourcing, e.g. Chittilappilly et al. [2016] focus on different topics, such as types of incentives, task recommendation and quality control systems.

A number of surveys have been conducted in the field of crowdsourcing [Chittilappilly et al. 2016; Mao et al. 2015a; Pan and Blevis 2011; Xintong et al. 2014; Yuen et al. 2011]. A short survey by Pan and Blevis [2011] presented a literature review of crowdsourcing and interaction design among academic, business, and social domains. This study was the first step for providing some insights and recommendations for designing crowdsourcing tasks and highlighted some challenges in task creation within Human-Computer Interaction (HCI). Yuen et al. [2011] showed different classification of crowdsourcing systems based on their applications, algorithms, performances and datasets. Xintong et al. [2014] presented the state of the art of using crowdsourcing in data mining. Mao et al. [2015a] conducted a survey on the use of crowdsourcing to support software engineering field.

Designing the task appropriately can lead to high-efficiency outcomes and a reduction in disagreements in the result [Garcia-Molina et al. 2016]. Catallo and Martinenghi [2017] define a taxonomy of designing crowdsourcing tasks based on four design dimensions inspired from the explicit control aspects of human computation mentioned in Law and Ahn [2011]. These dimensions are defined as *What* kind of task need to be solved, *Who* is going to solve it, *Why* the workers need to work on it, and *How* to process these tasks. This classification along with low levels components presented the main factors that involved in the process of designing crowdsourcing tasks.

Several studies considered that task design has a significant effect on the task outcomes. McDonnell et al. [2016] showed that designing the task in a way that reduces the cognitive load on workers...
significantly increases performances. Related to this, Yang et al. [2016] showed how task design properties are highly correlated with perceived task complexity.

Allahbakhsh et al. [2013] considered task design as one of the main dimensions that control the quality of the crowdsourcing system. Their proposed quality control approaches are the *design-time* approach, where the requester could use various techniques to control the quality of the task in the design stage; and the *run-time* approach, where requesters include some monitoring during the task running to prevent any mistakes or low-quality performance. These two approaches can help to control the quality of the result and can be applied separately or simultaneously on one task.

Moreover, the main aspect that should be considered is that the task will be done by a human not a machine, which is why the psychological aspects in designing the task should be analysed [Alonso 2013]. Deng et al. [2016] enumerates guidelines for workers, requesters, and platforms developers to enhance the services in crowdsourcing field. The study conducted survey instrument based on the worker’s experiences and how they interact with the crowdsourcing system. The aim of this study is to enhance the workplace quality by providing governance mechanisms to ensure transparency and fairness in the work environment.

There are several factors that affect the task design: the length of the task, the nature of the required work (for example writing, classifying, or designing), the use of training questions and examples, and also the graphical user interface which often varies according to the complexity of the task. This section will present some of these studies and highlight a number of essential factors that have been discussed in the last few years.

**Psychological Factors**

Human factors, such as psychological traits, are one of the main aspects that influence performance. Many researchers have studied the influence of personality traits which can have a positive or negative effect on the accuracy of different task design [Harrison et al. 2013; Kazai et al. 2011]. For example, distinguishing different visual designs for a task could trigger different emotion in the workers leading to variation in the results.

Kazai et al. [2011] analysed workers’ behaviours that lead to the classification of workers into five different types: *Diligent, Competent, Sloppy, Incompetent,* and *Spammers*. The authors tried to connect workers' characteristics and their personality traits with the accuracy and the average time for completing the tasks. The findings of this study showed that workers’ behaviours have a significant effect on accuracy for labeling tasks. On the other hand, the average time the workers spent in solving the task did not have a positive effect on the accuracy level for the task results for *Sloppy, Incompetent,* and *Spammer* workers. For connecting workers behaviours with the personality traits, *Conscientiousness* workers achieved a high level of accuracy.

Morris et al. [2012] looked at priming effects in micro-task crowdsourcing environments. They showed that priming workers could increase performances in creative tasks. While they show that priming has positive effects, they also note that it should be unconsciously provided to workers and that it does not substitute training done by means of instructions and examples. Consistently, in Harrison et al. [2013], they used emotion priming in visual judgments tasks. They pointed out that
while positive emotion has significant effects on the performance, negative emotions could also priming workers positively in some situations. Moreover, there are environmental priming that could affect workers differently and these beyond of control requesters.

André et al. [2014] looked at how groups of workers performed and showed that asking workers to contribute sequentially worked better than simultaneous collaboration in complex creative tasks. This finding proved the importance of crowdsourcing microtasks rather than sending it to a group of people to solve it together. The implications of this study point out that workers feel more secure when working independently. Several factors need to be investigated, such as the identity of the workers, the time of releasing the tasks online, and the nature of the tasks purposely motivated virtual teamwork among crowdsourcing systems.

Another study investigated interset tasks for image labelling; i.e. how workers are influenced by the type of task they have previously completed when working on a new task [Newell and Ruths 2016]. Moreover, the impact of any previous experience that the workers had and the rejection of a completed job have a significant effect on the workers' expectation for the upcoming work. McInnis et al. [2016] studied the impact of unfair job rejection on the workers and the subsequent risk. As a result of unfair rejections, workers tend to act safe and minimize the risk by accepting the same type of tasks or selecting a task from a limited number of requesters who have good repetition or previous successful business with them. This safe action could keep the workers safe from rejection, but it will also prevent workers from expanding their experience on the new type of tasks. Furthermore, the new requesters could face the risk of lack of turnout or have malicious workers only for their tasks.

Type of Tasks

An optimal task design for a crowdsourcing task might not translate well in a task of different nature. An analysis of the effect of different variables (e.g., interface, length, the number of items) on task completion performance have been shown not to generalise when the nature of the task is variable, e.g Marcus et al. [2012] has shown this effect for labeling (assign a label to an image) and counting (count the number of objects in an image) tasks.

Using closed questions, such as multiple choice answers, or using open question, such as providing a text box area for answers, has an influence on the workers performance. Some studies found that using predefined answers could save time and gain accurate answers [Jain et al. 2017]; whereas other studies found that these type of tasks could increase the number of malicious workers who complete all answers in the task quickly, in order to just gain rewards [Eickhoff and de Vries 2013; Gadiraju et al. 2015].

In addition, Moussawi and Koufaris [2013] highlighted that giving the workers some level of freedom in the way they perform and respond to the task leads to high motivation of the workers. Similarly, Eickhoff and de Vries [2013] state that the use of open questions could result in more creative answers and less cheating. In Eickhoff and de Vries’ [2013] study, it appears that using questions that require a text answer to get feedback is very helpful. The author conducted a number of repetitive tasks that handle large datasets incorporating some factors that could enhance the overall result. The first factor was forming the question correctly. He recommends that the question should asked in a simple and straightforward way so that it could be consistently understood by all workers. For questions with
labelling answers, he suggested replacing them with a numerical scale to prevent misunderstanding. Moreover, it is preferable to use a broad range of labels (with no more than 6-7 categories) to give the workers some level of flexibility in giving the right answer.

**Task Graphical User Interface (GUI)**

Since the task graphical interface is the main way for the workers to understand the job, it is fundamental to design adequate graphical user interfaces, that can help the workers understand the task requirements, the processes they need to follow, and the results expected from them. This process will have a strong impact on the worker’s performance. Allahbakhsh et al. [2013] define one of the factors that affect the quality of the outcomes as the user interface, which is the graphical design of the task: they found that implementing a simple interface could help the workers complete the task in a short time and increase the accuracy of the completed job. Furthermore, the study by Jain et al. [2017] showed that writing long instructions providing a detailed description of the task, and using examples, will have a positive effect on the quality of the result, particularly for complex tasks.

A study by Kim et al. [2015] used a crowdsourced task to match the appearance of the colour of some products on a website with the real colour of the same products. The lighting and the image quality that had been used in the task had a strong impact on the accuracy of the result. Other studies, such as Finnerty and Kucherbaev [2013] compared the outcomes of two tasks with simple and complex interfaces and the results proved that using a simple clear interface records higher result than using the same task contents, but with a complex interface (as defined in [Galitz 2007]).

Furthermore, a study by Alonso [2013] presented an interface design by following the guidelines of Nielsen [1993] to point out the basics of task design: write clear instructions, show examples, highlight and colour what is important and required for the job, which can reduce the effort to complete the task. Also, using a relevant, clear, and attractive title for the job will make it easy for workers to find it quickly when they are searching the platform for possible tasks to accept and complete.

McInnis et al. [2016] investigated a number of factors that lead to unfair rejection, such as insufficient task design, misleading instructions, technical errors, and requesters with poor knowledge. They concluded their study with a number of suggestions that could reduce the risk and enhance the connections between workers and requesters to achieve a better final outcome. One of these suggestions was to provide in the design of the task *an alarm* for a broken task, which notifies the requester of any error in of the task design during the work process.

Recently, Wu and Quinn [2017] outlined best practice guidelines for writing task instructions that could optimise the quality of the outcome. This study found that regardless of the facts that long and clear instructions will improve the result, workers tend to favour tasks with short guidelines and few lines of instructions. Therefore, the requesters should make a balance between presenting full instructions and defining attractive short steps which will be easy to read and deliver the full format of the task specification at the same time.

Gadiraju et al. [2017] also investigate the effect of the task clarity on the worker’s performance. Through surveying workers regarding their previous tasks and how often they found it clear, feedback
from workers pointed out that most of the features of an unclear task were because of the writing style and lack of detail in the presentation of the instructions. Also, they refer to the rare use of examples which make it less clear for them to understand the requirement of the job. The finding of this study showed that task clarity could be predicted and supervised via the proposed model and guide the requester in the task design. Further investigation could drown from this work to examine the relationship between task clarity and complexity and the effect of task clarity on workers dropout rates.

Recent study by Yang et al. [2016] proposed a high-dimensional regression model to measures the impact of task structural features on the complexity of the task and conversely using these features to predict the complexity and the tasks outcomes, showing that the semantic description and the visual appearance of the task are the most useful features to predict the complexity of the task and improve the quality of the output.

**Training Questions**

Training questions can be formed in a variety of ways which may be helpful for some tasks but not others. Several studies have looked at the training of workers before or whilst performing a particular task and a number of training techniques or methods were used. These can be summarised as follows:

1. *Control method*: does not have any training questions and the workers will read the instructions and start solving the task directly.
2. *Solution method*: adding a number of training tests before the real task questions without stating explicitly that first tasks are for training.
3. *Gold Standard*: the same setup as the solution method but after solving the first training tasks workers are shown the correct answers for the tasks and informed that they had been used for training purposes. Oleson et al. [2011] used this method in tasks as quality control mechanism rather than using it as a training method.
4. *Example method*: design task instructions to explain that workers will be shown some examples completed by an expert and that they are not allowed to start the task until the 30 second demonstration has been completed; this forces workers to read the examples and understand how they were solved. A recent study by Jain et al. [2017] and Wu and Quinn [2017] proved that using examples is crucial and plays a key factor in increasing the accuracy of results and the total agreement amongst workers. Similarly, Mitra et al. [2015], presented some examples for the workers followed by test questions to measure the improvement in their performance and to determine if they learned from the examples.
5. *Validation method*: in this method workers were shown two answers of other previous workers and asked to validate these answers by filling out some specific questions about them. Zhu et al. [2014] found that using the validation method in subjective tasks which required some creativity in devising the solution, was more effective than making the workers do more training tests.

Another study by Doroudi et al. [2016] presented five different ways of using training questions to improve the overall result of what they defined as a complex task. They used all five methods to find the most beneficial training method. The findings of this study reported that showing the workers
expert examples increased the overall accuracy of the answers compared with using other methods. Moreover, using the validation method was the most effective way of training workers.

**Length of the Microtask**

The length of the crowdsourced task can be designed to vary in length. To maintain a balance between the length of the task and the desired quality of the outcomes, several solutions have been proposed in different studies. One of these solutions is to decompose the long task into shorter ones (microtasks), which corresponds with the main goal of crowdsourcing platforms - to keep the tasks simple.

The main goal of using a crowdsourcing platform is to break down a task into smaller tasks - as we mentioned previously - which can be solved by the crowd, achieving high-quality performance as well as saving time and money [Cheng et al. 2015; Kittur et al. 2011]. These microtasks should have a low level of complexity to achieve their purpose. Doroudi et al. [2016], defined the level of complexity for a task: as a task which cannot be decomposed into micro-tasks and workers can use different mechanisms to perform such tasks. For complex task, a high level of accuracy is not achievable with low expertise workers.

Previous related work in the area of microtask crowdsourcing has looked at the effect on crowd performance of task granularity. For example, Cheng et al. [2015] showed that having shorter tasks lead to increased overall completion time but also to better quality contributions. Similarly, Allahbakhsh et al. [2013] discussed the granularity long tasks, which affects the quality of the outcomes. The final result of such a task is a combination of the results of a number of smaller or shorter tasks.

Another solution is to break the long task up with some activities to keep the worker interested in completing the task. Dai et al. [2015] proposed including some entertainment micro-tasks as a short break in performing a long task. They used the MTurk platform to design three different long tasks: (1) Classifying images, (2) Rating Wikipedia articles, and (3) Merging freebase entities. For each type of task, they inserted three different "micro-diversions": no diversion, a narrative webcomic story, and a dice game to keep workers on track and motivate them to continue working on the task. The findings of this study proved that using micro-diversions can significantly maintain workers' motivation to continue working on a long task as well as enhancing the speed of the answers. There are some variations in the findings depending on the task type and the micro-diversions combination. A complex cognitive task, such as rating a Wikipedia article, was performed more effectively using a diversions task. Moreover, the story acts better than a game diversion in speeding up workers' performance.

Moreover, Brambilla et al. [2015] propose prototyping methods for task design that will be implemented first in small datasets in order to gain better result for designing the same task for large datasets. This approach reports significant results for image relevance judgment tasks; further work could use the same strategies in other types of tasks.
Ordering Effects

In the process of implementing the task the ordering of data in the microtasks could lead to a variation on the overall result. The requester has the option to organise the data in the batch and present it in ascending order of difficulty that gradually prime the workers and improve their performance.

Cai et al. [2016] looked at how the sequence of writing tasks impacts crowd worker efficiency. They observed that by varying the order of task complexity and task type, workers' performance would vary thus indicating the potential to optimise worker efficiency by appropriately sorting tasks in a batch. Lasecki et al. [2015] looked at the effect of interruptions and of changing tasks type (i.e., context switch) on sequences of crowdsourcing tasks, showing how worker speed would significantly decrease in such situations.

Damessie and Culpepper [2016] investigated the impact on the inter-rater agreement of presenting documents in two different ways: (1) descending order of relevance, and (2) document identifier order where the documents vary depending on the topic. They designed a judgment task for 30 documents across easy and hard topics extracted from TREC collections and with a four-level relevance scale. The results showed that ordering by document identifier leads to a higher agreement in both easy and hard topics and a better result in term of identifying the relevant documents.
4 The Role of Microtask Work Environments

An aspect that has remained largely invisible in microtask crowdsourcing is that of work environments; defined as the hardware and software affordances at the disposal of crowd workers which are used to complete microtasks on crowdsourcing platforms. In [Gadiraju et al 2017], we reveal the significant role of work environments in the shaping of crowd work. First, through a pilot study surveying the good and bad experiences workers had with UI elements in crowd work, we revealed the typical issues workers face. Based on these findings, we then deployed over 100 distinct microtasks on CrowdFlower, addressing workers in India and USA in two identical batches. These tasks emulate the good and bad UI element designs that characterize crowdsourcing microtasks. We recorded hardware specifics such as CPU speed and device type, apart from software specifics including the browsers used to complete tasks, operating systems on the device, and other properties that define the work environments of crowd workers.

Our findings indicate that crowd workers are embedded in a variety of work environments which influence the quality of work produced. To confirm and validate our data-driven findings we then carried out semi-structured interviews with a sample of Indian and American crowd workers from this platform. Depending on the design of UI elements in microtasks, we found that some work environments support crowd workers more than others. Based on our overall findings resulting from all the three studies, we introduce ModOp, a tool that helps to design crowdsourcing microtasks that are suitable for diverse crowd work environments. We empirically show that the use of ModOp results in reducing the cognitive load of workers, thereby improving their user experience without affecting the accuracy or task completion time. An example of ModOp is shown in Figure 5 and described in the following.

![Fig. 5. Example of ModOp plugin in overlay over a task design.](image-url)
ModOp parses crowdsourcing tasks as HTML forms and guides a requester during the task design phase, by providing appropriate warnings and feedback according to our key findings. The elements that our tool monitors are:

- **Input Box / Text Area size** – Warning triggered if the size is disproportionately small or large.
- **Image size and resolution** – Warning triggered if the image size or resolution is disproportionately small.
- **Checkboxes** – Warning triggered if number of checkboxes is not optimal; requesters are advised to split checkbox questions where there are more than 10 options.
- **Radio Buttons** – Warning triggered if the number of radio buttons corresponding to a question is greater than 4.

Apart from the design feedback that is provided by ModOp on-the-fly, the tool can also supports requesters in making work environment-aware decisions during task design:

- **Device Type** – ModOp automatically detects the device type of workers by leveraging the user agent string.
- **Device Speed** – ModOp automatically provides an estimate of the worker’s device speed based on a target relative speed.
- **Screen Resolution** – ModOp automatically detects the screen resolution of the worker’s device.

With minimal effort, requesters can integrate ModOp into their task design workflow and make informed work environment-aware decisions; this can facilitate more inclusive pay schemes (for example, awarding bonuses to workers for good performance despite poor work environments), shape task assignment and routing (for example, routing tasks that contain high resolution media to workers with large Internet connection bandwidths and fast devices), and have broad implications on fairness and transparency in crowd work.

We believe that this tool can help crowdsourcing requesters in designing better tasks by directing requesters’ attention to otherwise neglected attributes of UI element design and work environments. We envision that this would improve the overall crowd worker experience and help in reducing their cognitive load.
5 Crowdsourcing Interfaces in Fashion

In WP3 we focus on entity linking: given a real world image we use crowdsourcing to link all the fashion products that are visible in the image to an existing catalog. In the optimal case the crowd should link the “perfect match” (the exact same product) from a predefined set of products. In case the “perfect match” is not present in the database or the crowd does not find the optimal solution the most similar product should be annotated.

Task Description

Product Linking Task:

The Product Linking Task has the following flow:

1. Each worker is presented with an independent task, no collaboration allowed (as suggested by André et al. [2014]).
2. Each independent task has one image that represents one or multiple products.
3. The worker has to signal (optionally) if this record has a data error (missing image etc) or if the images does not contain fashion products.
4. Each product can be considered as a microtask
   a. The worker is presented with an image patch containing a single product and a sublist (16) of suggested products out of the entire product database.
   b. The workers’ task is to find the “perfect match” out of the entire product database.
   c. In case the sublist does not contain the “perfect match” the worker has the following options to modify the list: text search and “iterative image search” (see figure 8 for a visual example)
   d. If no perfect match can be found, the worker should select the most similar product.
   e. Once a product is selected, the worker is signaling the quality of the match with a star rating (1 to 3).
5. After all microtask have been done, the worker clicks on the “save” button and the next task is presented to the worker.

The structure of the possible worker response (predefined options followed by text and image search) is carefully selected to minimise the worker load when a creative solution is not needed (Jain et al. [2017]) and providing a level of freedom when needed, increasing workers motivation (Moussawi and Koufaris [2013], Eickhoff and de Vries [2013]).

Task Instructions

Each worker receives one hour training (via video conference and screen sharing) before the beginning of the job: this has been proven to have a positive effect on the quality of the result, in accordance to the study of Jain et al. [2017] for complex tasks. The training consisted of a multiple run of the task via examples (Alonso [2013]). Moreover, the worker is able to continuously communicate with the requester via instant messaging, receiving instantaneous feedback and clarification in case of doubts.
Task Graphical User Interface (GUI)

An overview of the Task GUI is shown in Figure 6:

- The task GUI consists of header (top part) and body (bottom part).
- The header presents the worker with an image and some additional information (e.g. who created the image in case the image comes from a social media post, and some text description that accompanies the image).
- The body presents the worker with micro-tasks, i.e. with image patches for each product found in the header image.

Fig. 6. Overview of Task Graphical User Interface.
An overview of the row structure is shown in Figure 7. Each row (micro-task) consists of:

- The image patch (original image) of a product
- Info box which presents the worker with information of gender and fashion category (e.g. jumpers, pants, sunglasses, etc.)
- An image of a matched candidate from the product database
- Product suggestions which allow the worker to select the “perfect match” product
- Text search field through which product suggestions list can be refined by fashion category, description, brand
- Match rating which the worker uses to rate how good the match is, ranging from 1 to 3 stars, higher is better (i.e. in case of the “perfect match” the worker would select 3 stars).

Fig. 7. Overview of the row structure.
The interaction with each row is simple and consists of the following steps:
1. The worker is first presented with product suggestions for the **original image** (top part).
2. Then the worker can select a similarly looking product from the product suggestions list.
4. In case the worker is already happy with the suggested product he can already rate the suggestion.
5. In case the worker is not yet happy with it he can refine further until he is confident enough to rate the suggestion. There are several options for refinement:
   
   **Iterative Visual search:** After that, product suggestion list is refreshed with products similar to the **matched product** (bottom part), thus allowing user to move in space of similar fashion product until he/she gets to the most similar one.
   
   **Text Search:** The worker can also type in keywords in the search-bar and the suggested product change accordingly.

![Fig. 8. Overview of the iterative visual search.](image-url)
An overview of the final rating is shown in Figure 9: in the end of the task the worker rates the match, e.g. in the example shown in the figure the jumpers are quite similar but not a “perfect match”, so the worker should rate it with 2 stars.

![Image of jumpers with rating 3 stars]

Figure 9. Overview of final rating system.

Strategies on GUI design

The following guidelines has been used to maximise the quality:

- We focused on designing simple and clear interfaces (Finnerty and Kucherbaev [2013], Nielsen [1993], Gadiraju et al. [2017]).
- We minimise the risk of unfair rejection and worker dissatisfaction by providing continuous feedback with clear explanation of potential mistakes (McInnis et al. [2016]).

Training

- The training of the workers is important for the success of the overall task. The workers have to be highly skilled for this task, because they need to have a deep fashion knowledge in order to point out the difference between two different products.
- The learning curve is steep. We noticed that trained and skilled workers have a much higher throughput (up to 3-4 times) with significant higher quality. Speed and quality increase once they understand the performance of the ‘iterative image search” and once the worker get a good feeling for the product catalog.
- The training of the worker is done using video conference and screen sharing.
- The trainer starts with 5 standard task, so that the worker gets familiar with the task.
- The next step is that the worker performance 2-3 task and the trainer helps in that process.
- Once the worker is familiar with the standard task, the trainer shows some borderline cases.
- After the first day the trainer reviews random task and gives direct feedback. The same process is performed several times throughout the month in order to ensure the quality of the worker.
Fig. 8. Task Infrastructure. In order to provide the ‘suggestions’ and the ‘visual similar’ product feature several components needed to be implemented. The product detector runs over the image and localizes if a product is visible. The Visual Index performs the ‘similarity search’ for a given image patch, this means it finds the most similar product out of our product database.

5 Conclusion

We have reviewed the literature on task design within crowdsourcing with an emphasis on user interface design and worker training. We then present interface designs for the problem of entity linking and images in fashion. The resulting process is complex, and requires a different approach of the traditional crowdsourcing methods: in particular we employed a continuous two-way feedback system with the workers, and an extensive initial training of one hour per worker, which have been proven being very effective in obtaining the expertise required for such a complex task. Many of the results in the literature regarding task clarity, micro-task segmentation and interface design have been confirmed by our experiments.


