Refining Linked Data with Games with a Purpose: Postprint

Authors: Celino, Irene, Calegari, Gloria Re, Fiano, Andrea, Celino, Irene

Date: 2022-11-18T00:00:00+00:00

Abstract

With the rise of linked data and knowledge graphs, the need becomes compelling to find suitable solutions to increase the coverage and correctness of data sets, to add missing knowledge and to identify and remove errors. Several approaches – mostly relying on machine learning and natural language processing techniques –have been proposed to address this refinement goal; they usually need a partial gold standard, i.e., some "ground truth" to train automatic models. Gold standards are manually constructed, either by involving domain experts or by adopting crowdsourcing and human computation solutions. In this paper, we present an open source software framework to build Games with a Purpose for linked data refinement, i.e., Web applications to crowdsource partial ground truth, by motivating user participation through fun incentive. We detail the impact of this new resource by explaining the specific data linking "purposes" supported by the framework (creation, ranking and validation of links) and by defining the respective crowdsourcing tasks to achieve those goals. We also introduce our approach for incremental truth inference over the contributions provided by players of Games with a Purpose (also abbreviated as GWAP): we motivate the need for such a method with the specificity of GWAP vs. traditional crowdsourcing; we explain and formalize the proposed process, explain its positive consequences and illustrate the results of an experimental comparison with state#2;of-the-art approaches. To show this resource' s versatility, we describe a set of diverse applications that we built on top of it; to demonstrate its reusability and extensibility potential, we provide references to detailed documentation, including an entire tutorial which in a few hours guides new adopters to customize and adapt the framework to a new use case.

Full Text

Preamble

Refining Linked Data with Games with a Purpose

Irene Celino†, Gloria Re Calegari & Andrea Fiano
Cefriel, Milano 20126, Italy

Keywords: Human computation; Games with a purpose; Linked data; Knowledge graph; Data refinement; Data linking; Truth inference

Citation: I. Celino, G. Re Calegari & A. Fiano. Refining linked data with games with a purpose. Data Intelligence 2(2020), 417-442. doi: 10.1162/dint_a_{00056}

Received: April 15, 2019; **Revised:** August 10, 2019; **Accepted:** October 10, 2019

ABSTRACT

With the rise of linked data and knowledge graphs, finding suitable solutions to increase dataset coverage and correctness, add missing knowledge, and identify and remove errors has become increasingly compelling. While several approaches—mostly relying on machine learning and natural language processing—have been proposed to address this refinement goal, they typically require a partial gold standard, or "ground truth," to train automatic models. Such gold standards are manually constructed either by domain experts or through crowdsourcing and human computation solutions. In this paper, we present an open-source software framework for building Games with a Purpose (GWAP) for linked data refinement. These web applications crowdsource partial ground truth by motivating user participation through fun incentives. We detail the impact of this resource by explaining the specific data linking "purposes" supported by the framework (creation, ranking, and validation of links) and defining the respective crowdsourcing tasks to achieve these goals. We also introduce our approach for incremental truth inference over contributions provided by GWAP players. We motivate the need for such a method given the unique characteristics of GWAPs versus traditional crowdsourcing, explain and formalize the proposed process, discuss its positive consequences, and illustrate the results of an experimental comparison with state-of-the-art approaches. To demonstrate the framework' s versatility, we describe a set of diverse applications built on top of it. To show its reusability and extensibility potential, we provide references to detailed documentation, including a complete tutorial that guides new adopters through customizing and adapting the framework to a new use case in just a few hours. The framework code and tutorial are available on GitHub and documented on Apiary (links provided throughout the paper).

† Corresponding author: Irene Celino (Email: irene.celino@cefriel.com; ORCID:

0000-0001-9962-7193)

*Extended version of the paper "A Framework to Build Games with a Purpose for Linked Data Refinement," accepted at the 17th International Semantic Web Conference.

---

1. INTRODUCTION

In the era of data-driven technologies, evaluating and increasing data quality has become critically important. Within the Semantic Web community, the emergence of knowledge graphs has been celebrated as a success of the linked data movement, yet it has simultaneously raised numerous research challenges regarding their management—from creation to verification and correction. The term "knowledge graph refinement" [1] refers to the process of improving knowledge graph quality by finding and removing errors or adding missing knowledge. This same refinement operation poses a challenge for any linked dataset of considerable size.

Addressing the linked data refinement problem at scale requires a trade-off between purely manual and purely automatic methods. Effective approaches adopt human computation [2] and crowdsourcing [3] to collect manually labeled training data on one hand, while employing statistical or machine learning techniques to build models and apply refinement operations at larger scales on the other. Indeed, active learning [4] and other recent research in artificial intelligence have reintroduced the "human-in-the-loop" by creating mixed human-machine approaches.

This paper presents an open-source, reusable resource for building human computing applications in the form of Games with a Purpose [5] aimed at solving linked data refinement tasks. The framework can be adopted both for refining a pre-existing knowledge graph (when human intervention alone suffices to complete the refinement task) and for building a gold standard (to be used as a training set when machine learning approaches are needed to process very large linked datasets). In this paper, "refinement" primarily refers to data linking, as more precisely defined in Section 3. The presented resource consists of both a software framework and a crowdsourcing approach that can be customized and extended to address different data linking issues.

The paper is organized as follows: Section 2 presents related work; Section 3 defines the refinement purpose addressed by our framework; Section 4 explains the crowdsourcing task; details about our incremental truth inference algorithm are given in Section 5; the software resource is presented in Section 6; applications built on it are illustrated in Section 7; an evaluation of the truth inference approach' s effectiveness appears in Section 8; since this is a recently released resource, we have prepared a complete tutorial to explain its potential customization and simplify adoption, briefly introduced in Section 9; the frame-

work code and tutorial are available on GitHub and documented on Apiary (links throughout the paper); Section 10 concludes.

2. RELATED WORK

Data linking is rooted in the record linkage problem studied in the databases community since the 1960s [6]. In the Semantic Web community, the term often refers to finding equivalent resources on the Web of linked data [7]—the process of creating links that connect subject and object resources from two different datasets through a property indicating correspondence or equivalence (e.g., `owl:sameAs`).

We generalize the concept of data linking to the task of creating links in the form of RDF triples, without limiting specific types of resources or predicates, nor necessarily requiring linking across different datasets or knowledge graphs (data linking can also occur within a single dataset or knowledge graph). In this sense, data linking serves as a solution to linked data refinement issues—the process of creating, updating, or correcting links in a dataset, where a link is any subject-predicate-object triple relation. As defined in [1] with respect to knowledge graphs, data linking does not involve constructing a dataset from scratch but rather assumes an existing input dataset that needs improvement by adding missing knowledge or identifying and correcting mistakes.

The Semantic Web community has long investigated methods to address data linking, identifying linked dataset quality assessment methodologies [8] and proposing manual, semi-automatic, or automatic tools for refinement operations [9, 10]. The majority of refinement approaches, particularly for knowledge graphs requiring scalable solutions, employ various statistical and machine learning techniques [11, 12, 13, 14].

However, machine learning methods require a partial gold standard to train automated models. These training sets are usually created manually by experts —a process that, while typically yielding higher-quality trained models, is also expensive, resulting in relatively small "ground truth" datasets. Involving humans at scale in an effective manner is the goal of crowdsourcing [3] and human computation [2]. Indeed, microtask workers have been employed to perform manual quality assessment of linked data [15, 16].

Among different human computation approaches, Games with a Purpose (GWAP) [5] have achieved wide success due to their ability to engage users through fun incentives. A GWAP is a gaming application that leverages players' actions to solve hidden tasks. Users play for enjoyment, but the collateral effect of their gameplay is that comparing and aggregating players' contributions solves problems—usually labeling, classification, ranking, clustering, among others. The Semantic Web community has adopted GWAPs to address numerous linked data management issues [17], from multimedia labeling to ontology

alignment, error detection to link ranking: data cleansing in DBpedia [18], collecting linguistic annotations [19], ontology alignment [20], rating triples based on popularity [21], linking multimedia datasets about smart cities [22], crowdsourcing location-based knowledge [23], annotating art-related media to improve search engines [24], evaluating people' s ability to recognize and build triples [25], ranking artworks by recognizability while creating awareness [26], and building training sets for image classification [27].

Aggregating user contributions is a key issue in human computation systems like GWAPs. Originally, aggregation relied on simple agreement: in the ESP game [28], the very first GWAP, players typed textual labels to tag images, and two agreeing users were enough to consider the label "true." Afterwards, "ground truth" tasks—problems with known solutions—were introduced to check contribution quality and cope with random answers or malicious players [2, 29].

In crowdsourcing settings, truth inference is typically computed ex-post, meaning contribution aggregation occurs only after collecting all data from all users [30]; the number of repetitions (i.e., different users required to solve the same task) is set at design time. Related research has investigated repeated labeling strategies [31] to understand when collecting user contributions becomes redundant. Promising results for optimizing the number and quality of collected contributions come from active learning research [32, 33], which integrates machine learning with crowdsourcing approaches, even for large-scale datasets. This paper introduces our approach for incremental truth inference that addresses minimization of repeated labeling by dynamically detecting when "enough" repetitions have been collected.

While general guidelines for building GWAPs have been described and formalized [34] (including game mechanics like double-player mode, answer aggregation through output agreement, task design, among others), building a GWAP for linked data management still requires time and effort. To our knowledge, source code has only been made available for the labeling game Artigo [24].

---

3. DATA LINKING PURPOSE

As explained in Section 2, data linking refers to the general problem of creating links in the form of triples. This section provides basic definitions and illustrates the cases our framework supports as purposes for games built on it.

3.1 Basic Definitions

The following formal definitions are used throughout the paper.

Resources R is the set of all resources (and literals), whenever possible also described by their respective types. More specifically: $R = R_s \cup R_o$, where R_s is the set of resources that can take the role of subject in a triple and R_o is the

set of resources that can take the role of object in a triple. The two sets are not necessarily disjoint, i.e., $R_s \cap R_o \neq \emptyset$.

Predicates P is the set of all predicates, whenever possible also described by their respective domain and range.

Links L is the set of all links. Since links are triples created between resources and predicates, $L \subset R_s \times P \times R_o$. Each link is defined as $l = (r_s, p, r_o) \in L$ with $r_s \in R_s$, $p \in P$, $r_o \in R_o$. L is usually smaller than the full Cartesian product of R_s, P, R_o because in each link (r_s, p, r_o) it must be true that $r_s \in \text{domain}(p)$ and $r_o \in \text{range}(p)$.

Link scores s is the score of a link—a value indicating confidence in the link' s truth value, usually $s \in [0, 1]$. Each link $l \in L$ can have an associated score.

One final note on subject resources: in the following sections and in the framework implementation, we always assume that any subject entity can be shown to players through some visual representation. If the entity is a multimedia element (image, video, audio resource), this requirement is automatically satisfied. In other cases, additional information may be required: e.g., a place could be represented on a map, a person through their photo, or a document through its textual content.

3.2 Data Linking Cases

Given these definitions, we can split the general data linking problem into more specific cases:

Link creation A link l is created: given $R = R_s \cup R_o$ and P, the link $l = (r_s, p, r_o)$, $r_s \in R_s$, $p \in P$, $r_o \in R_o$ is created and added to L. All three link components exist—they are already included in sets R and P.

Classification can be seen as a special case of link creation: given a resource $r_s \in R_s$ to be classified and predicate $p \in P$ indicating the relation between the resource and a set of possible categories $\{cat_1, cat_2, \dots, cat_n\} \subset R_o$, the resource $r_o \in \{cat_1, cat_2, \dots, cat_n\}$ is selected to create link $l = (r_s, p, r_o)$. For example, in music classification: given a list of resources of type "music tracks" in R_s, predicate `mo:genre \in P`, and a set of musical styles in R_o, the task is to assign a music style to each track by creating the link $(track, genre, style)$.

The framework presented here supports any link creation case (including classification) when resources and predicates are already part of R and P. Link creation where new resources and/or predicates are added to R and/or P (e.g., free-text labeling of images) is currently not supported but could be a possible extension.

Link ranking Given the set of links L, a score $s \in [0, 1]$ is assigned to each link l. The score represents the probability of the link being recognized as true. Links can be ordered by score s, yielding a ranking.

We consider a Bernoulli trial where the experiment consists of evaluating a link' s "recognizability," with "success" when the link is recognized and "failure" when not. Under the hypothesis that the success probability remains constant across trials, link score s estimates the binomial proportion in the Bernoulli trial.

In human computation, crowdsourcing, or citizen science, each trial consists of a human user evaluating the link and stating whether, in their opinion, the link is true (success) or false (failure). If suitably selected, human evaluators can be considered a random sample of the entire human population. Aggregating sample results thus estimates a link' s truth value for the entire population by computing the probability of each link being recognized as true. Ordering links by score provides a metric to compare different links on their respective "recognizability."

For example, ranking photos depicting a specific person (e.g., an actor, singer, or politician): given a set of images, human-based evaluation could identify pictures where the person is most recognizable or clearly depicted.

Link validation Given the set of links L, a score $s \in [0, 1]$ is assigned to each link l. The score represents the link' s actual truth value. A threshold $\bar{s} \in [0, 1]$ is set so all links with score $s \geq \bar{s}$ are considered true.

Link validation differs from link ranking in two ways: first, each link is considered separately in validation, while ranking aims to compare links; second, while human judgment estimates subjective population opinion in ranking, validation assumes that if a link is recognized as true by the human population (or a sample), this is a good estimation of the link' s actual truth value. This also motivates the threshold $\bar{s}$: while truth values are binary (0=false, 1=true), human validation is fuzzier with "blurry" boundaries. The optimal threshold is domain- and application-dependent and usually estimated empirically.

An example would be assessing correct music style in audio tracks: music genres sometimes overlap, and identification can be subjective (e.g., there is no strict definition of "rock"). Employing humans in validation means attributing the most shared evaluation of a music track' s genre.

As mentioned, human evaluation of a link can be considered a Bernoulli trial: each link l is assessed n times by n different users u; the link is recognized as true X times (with $X \leq n$). Each user u_i may be more or less reliable, and in some cases their reliability r_i can be estimated. Therefore, link score is $s = f(n, X, r)$ —a function of trial count n, success count X, and involved users' reliability values $r = \{r_1, r_2, \ldots, r_n\}$.

---

4. CROWDSOURCING TASKS FOR DATA LINKING

Our framework enables designing and developing GWAP applications to solve data linking problems. Games built on our framework ask players to solve

atomic data linking issues as basic tasks within gameplay.

Building a GWAP does not automatically guarantee collecting enough players/games to solve the data linking problem at hand. However, our experience shows that if the task is properly embedded in simple game mechanics and targeted to a specific community of interest, a GWAP is a valuable means to collect a "ground truth" dataset for training machine learning algorithms [27].

4.1 Game Basics

Each GWAP built with our framework is a simple casual game organized in rounds. Each round consists of several levels, and each level requires the player to perform a single action corresponding to link creation, ranking, or validation. Following von Ahn's definition [17], each GWAP is an output-agreement double-player game: users play in random pairs, and game scoring is based on agreement between players' answers (i.e., they gain points if they agree). Our framework does not require simultaneous play, implementing a common strategy where a user plays with a "recorded player," so scoring is obtained by matching answers provided at different times.

The framework supports both time-limited rounds (where players answer a variable number of tasks depending on speed) and level-limited rounds (with a maximum number of tasks per round). The choice depends on task difficulty and game incentive mechanisms.

The game adopts a repeated linking approach by asking different players to address the same data linking task, though the same task is never given twice to the same player. A data linking task's "true" solution therefore emerges from aggregating answers from all users who "played" the task in any round. The number of players required depends on the aggregation algorithm, as explained in Section 4.3.

Notably, game scoring (points gained by players) is not directly related to data linking scoring (attributing score s to link l): the former is an engagement mechanism to retain players, while the latter is the game's actual purpose.

4.2 Crowdsourcing Definitions

We formulate the crowdsourcing problem in a GWAP with the following definitions. For simplicity, we specifically consider multinomial classification tasks (with a pre-defined label set), though the approach extends easily to open labeling or other data linking cases without loss of generality.

Consider a Game with a Purpose aimed at solving a set of tasks $T = \{t_n | n = 1, 2, \dots, N\}$, say, classifying painting images. Each task is a labeling task assigning a label from admissible values $V = \{v_l | l = 1, 2, \dots, L\}$, say, painting techniques (e.g., watercolor, gouache, oil paint).

The GWAP is played by users $U = \{u_k | k = 1, 2, \dots, K\}$. In each round, a player

is assigned a subset $T' \subset T$ of tasks, e.g., a set of painting pictures to classify. Given "ground truth" tasks $G = \{g_m | m = 1, 2, \ldots, M\}$ with known solutions (e.g., paintings with known techniques), each round also gives the player control tasks $G' \subset G$. Control tasks appear identical to unsolved tasks so users cannot distinguish them. Answers to control tasks estimate player reliability q_k, useful for "weighting" contributions on unsolved tasks during truth inference. If player 9 correctly answers 3 out of 4 control tasks, their reliability for that round is $q_9 = 0.75$.

Player contributions are collected in matrix $C = \{c_{n,k} | n = 1, 2, \ldots, N \wedge k = 1, 2, \ldots, K\}$, initialized with null/zero values and filled with labels from V when a player completes a task (e.g., if player 3 says painting 5 is watercolor, $c_{3,5}$ becomes "watercolor"). The GWAP' s goal is not to completely fill C; rather, C should remain sparse, with the minimum possible number of player contributions (non-zero values) required to infer tasks' "true" labels.

Truth inference is a function applied to players' answers C and reliability values q_k to infer result set $\hat{Y} = \{\hat{y}_n | n = 1, 2, ..., N\}$, with $\hat{y}_n \in V$ for each task in T. $\hat{Y}$ is computed by aggregating user contributions and estimates the unknown "true" labeling Y of tasks (e.g., computing the "true" painting technique for each image). Truth inference is incremental if each new GWAP player contribution triggers a new $\hat{Y}$ estimation.

To determine whether task t_i can be considered completed, truth inference computes scores representing confidence values for associations between t_i (a painting) and each possible labeling value v_i (a painting technique). The aggregation algorithm builds and updates an estimation score matrix $S = \{s_{n,l} | n = 1, 2, \ldots, N \wedge l = 1, 2, \ldots, L\}$, where $s_{n,l} \in [0, 1]$. As in record linkage literature [11], these scores start at 0 and incrementally increase according to user contributions. Task t_i is solved when the maximum of its scores $s_{i,*}$ (the ith row of matrix S) exceeds threshold $\bar{s}$. The completion condition is:

$$\forall i \in \{1, 2, \ldots, N\}, \exists j \in \{1, 2, \ldots, L\} : s_{i,j} \geq \bar{s}$$

For example, if threshold $\bar{s} = 0.80$ and painting 7's scores are $s_{7,\text{watercolour}} = 0.25$, $s_{7,\text{gouache}} = 0.31$, and $s_{7,\text{oil}} = 0.82$, the algorithm assigns "oil" painting technique to painting 7 and the task is completed. Truth inference algorithms differ in their specific approaches to updating score matrix S when aggregating user contributions.

4.3 Atomic Tasks and Truth Inference

As mentioned in Section 4.1, the atomic task in any GWAP built with our framework is an individual data linking task. For example, in music classification, a player might receive an audio track (resource r_s), relation `mo:genre` (predicate p), and options for music genres (e.g., classical, pop, rock, electronic, representing potential objects r_o of link l). The player' s action is to choose the genre

(resource r_o, say 'rock') that best describes the audio track. By performing this atomic task, the player indicates belief that link $l = (r_s, p, r_o)$ is "true" ; the game then alters link l' s truth score s by incrementing it. The score is modified at every player action.

Truth inference algorithms [30] have been heavily explored in crowdsourcing to aggregate and interpret workers' contributions. Most state-of-the-art algorithms (e.g., majority voting, expectation maximization [35], message passing [36]) are computed ex-post: all contributions are first collected, then aggregated, usually through iterative algorithms that infer truth and estimate worker quality until convergence. This requires setting a priori the number of repetitions for user labeling on each task, possibly collecting redundant information.

In most crowdsourcing platforms (like Amazon Mechanical Turk or Figure Eight), tasks are assigned in batches or Human Intelligence Tasks (HITs), and workers must submit answers within a specific timeframe to receive payment [37]. In contrast, GWAPs collect contributions as soon as users decide to play. The incoming answer flow depends on player appreciation of the game, often showing a long-tail effect: few players play many rounds while most participate for only a few minutes. Therefore, exploiting every single player' s contribution and inferring truth incrementally—assigning the same task to the minimum sufficient number of different players—is paramount.

To address this requirement, our GWAP Enabler framework implements truth inference as illustrated in Figure 1 [Figure 1: see original paper]. Each time a player starts a round, we assign a set of tasks, some of which are control tasks. We collect answers and compute player reliability. Then, for each unsolved task, we perform a truth inference step, incrementally computing a new task solution estimate. If the new estimation is "good enough" (per Equation (1)' s exit condition), the task is considered solved, removed from the game, and its result returned. Otherwise, the task remains in the game for the next user/player.

The next section explains the incremental truth inference algorithm and its details. Section 8 reports benefits of this approach with experimental evaluation.

---

5. INCREMENTAL TRUTH INFERENCE ALGORITHM

This section details our incremental truth inference approach: main requirements, pseudo-code algorithm, and discussion of prerequisite satisfaction.

5.1 Requirements

For Games with a Purpose, specific requirements motivate a new truth inference approach:

[R1] Dynamic estimate of labeling quality by computing player reliability on control tasks. Quality estimation is standard in crowdsourcing, but microtask workers solve all assigned tasks at once. We must account for GWAP players participating at different moments with varying attention levels, so their quality/reliability can change over time and cannot be computed once and for all.

[R2] Coping with varying task difficulty, including possibly multiple or uncertain classification tasks, meaning we cannot make a priori hypotheses about the number of redundant labeling actions required per task.

[R3] Incremental computation of truth inference: as introduced in Section 1, we want to aggregate contributions as soon as they are available because GWAPs have no predefined timeframe for player input.

[R4] Dynamic minimization of required repeated labeling to avoid useless redundancy: if a task is "easy" (i.e., not controversial or not requiring specific expertise), we want to automatically ask fewer players to solve it, while "hard" tasks should automatically remain in the game longer to be "played" by more users.

5.2 Algorithm

The approach outlined in Figure 1 is detailed in Algorithm 1. Each time a player starts a game round (line 2), they are assigned a set of link refinement tasks.

The player provides answers without distinguishing unsolved tasks from control tasks (cf. lines 6 and 14). Control task answers compute player reliability as a function of mistake count (lines 5-11); reliability is computed per game round.

The `ComputeUserReliability` function (line 11) can be implemented variously: simplest is using the percentage of correct labels in control tasks, i.e., $r_k \leftarrow 1 - \text{errors}/\text{size}(G')$. In some cases, it may be safer to strongly penalize players submitting random answers. In games built with our framework (cf. Section 7), we adopt a conservative estimation using:

$$r_k \leftarrow e^{-a \cdot \text{errors}}$$

where a is set so that r_k almost halves with 1 mistake and quickly decreases with further errors.

Answers on unsolved tasks are weighted by reliability and used to update estimation scores (lines 14-15). For each task t_i and possible solution v_j, the `UpdateSolutionEstimate` function implements:

$$s_{i,j} \leftarrow s_{i,j} + d \cdot r_k \quad \text{if } c_{i,k} = v_j$$

where $c_{i,k}$ is user u_k' s answer with reliability r_k and d is an increment depending on minimum required redundancy. In other words, the score of the user-indicated link increments (weighted by user reliability), while other links' scores remain unchanged.

At each truth inference step, task completion is checked (line 16) using Equation (1). If satisfied, the task solution is returned and the task removed from the game (lines 17-18). The algorithm iterates until all tasks are solved (line 1) and truth is inferred on all tasks (line 22).

5.3 Requirement Satisfaction

We qualitatively assess how our approach addresses Section 5.1' s requirements and discuss positive consequences.

Labeling quality is controlled via estimation score updates $s_{n,l}$, incremented with player contributions weighted by reliability values r_k. This means the approach considers contribution quality and "measures" it at each gameplay, relying on a "local" trustworthiness value. Dynamic re-computation of r_k fulfills requirement [R1], addressing the fact that the same player may show different behavior at different moments—e.g., being careful versus distracted.

The estimation scores $s_{n,l}$, their update function (cf. Equation (3)), and task completion condition (cf. Equation (1)) have additional interesting properties. Scores are attributed to each task-label combination and updated at each user contribution.

If task t_i is "easy," different players will attribute the same label v_j and the respective score $s_{i,l}$ will quickly increase and exceed threshold $\bar{s}$. Conversely, if a labeling task is difficult or controversial, different GWAP players may give different solutions from set V for the same task t_i, so potentially all scores in $s_{i,*}$ get updated but none easily exceeds $\bar{s}$.

Thus, the approach fulfills requirement [R2] on task difficulty (easy and difficult tasks are automatically detected and treated accordingly) and [R4] on repeated labeling (the number of players asked to solve the same task is dynamically adjusted).

Note that in record linkage literature [6], scores are assigned to each possible record pair, and usually the "matching" score increases while "non-matching" scores decrease respectively. For possibly multiple labeling and uncertain solutions (cf. requirement [R2]), we propose increasing the user-provided solution' s score without decreasing alternative solutions' scores. Variations of Equation (3)' s update function can be introduced depending on scenario characteristics. For example, if $c_{i,k} \neq v_j$, then $s_{i,j}$ could be decreased by quantity $d' \cdot r_k$, where d' is the decrement amount.

By design, Algorithm 1 fulfills requirement [R3], as each player contribution (line 14) triggers a truth inference step (line 15) and exit condition check (line 16).

This incremental approach ensures tasks are assigned to players only until an inferred "true" solution is reached, avoiding useless labeling redundancy (again satisfying requirement [R4]).

Dynamically adjusted repeated labeling also indirectly estimates task complexity: the more contributions needed to satisfy Equation (1)' s exit condition, the more difficult the task. Therefore, when task difficulty assessment is required, the number of collected contributions can serve as a proxy measure. Our previous work [27] demonstrated that this empirical difficulty measure highly correlates with (lack of) confidence values from machine learning classifiers applied to the same data.

A final note on task assignment: it is common best practice to give each crowd worker a task at most once and perform answer aggregation on responses from different workers. This also applies to GWAPs, as the same player could get bored if asked to solve the same problem repeatedly. Therefore, task assignment to player u_k (lines 3 and 12) takes tasks from G and T among those u_k never solved before. A pragmatic strategy to avoid exhausting the entire control task set G—which we typically adopt when implementing GWAPs—is to dynamically increment G by adding solved tasks from T (those removed when "true" solution is inferred), so line 18 could become: $T \leftarrow T - \{t_i\}; G \leftarrow G + \{t_i\}$.

---

6. THE GWAP ENABLER FRAMEWORK

To better illustrate our software framework, we provide technical details about its internals. The GWAP Enabler is released as open source under Apache 2.0 license and is available at https://github.com/STARS4ALL/gwap-enabler.

6.1 Structure of the Framework

Our GWAP Enabler is a template web application formed by basic software building blocks providing core functionalities for building linked data refinement games. By customizing this "template," any specific GWAP application can be built. The three main components are the User Interface (UI), Application Programming Interface (API), and Database (DB).

The main database table is `resource_{has}_{topic}`, containing all links $l = (r_s, p, r_o) \in L$. Each link has a score $s \in [0, 1]$ updated every time a user plays it. Link subject r_s and object r_o resources are stored in tables named `resource` and `topic`, respectively. For example, in the music classification case from Sections 3 and 4, the `resources` table would contain audio tracks and the `topics` table would list possible music styles. These three tables contain all data linking problem information and are initially filled according to the specific GWAP' s purpose.

Additional tables customize game behavior: the `configuration` table changes truth inference parameters, and the `badge` table modifies badges given as re-

wards. Remaining tables manage internal game data such as users, rounds, leaderboards, or logs, and are automatically filled during gameplay. These don' t require modification but can be freely adapted by developers, with awareness that altering them requires code modifications.

Technically, the GWAP Enabler consists of an HTML5, AngularJS, and Bootstrap frontend; a PHP backend managing application logic; and a MySQL database, as shown in Figure 2 [Figure 2: see original paper]. Frontend-backend communication occurs through an internal web-based API whose main operations retrieve links to show players in each round, update link scores based on player agreement/disagreement, and update player points for leaderboards and badges.

An external web-based API is also set up to feed the game with new data and access GWAP results aggregated by the truth inference algorithm. API methods allow submitting new tasks, retrieving solved tasks (= refined links) in JSON and JSON-LD formats, getting the number of tasks remaining, and listing key GWAP KPIs (e.g., throughput, average life-play or ALP, and expected contribution [17]) for game evaluation. This API is exhaustively documented at https://gwapenablerapi.docs.apiary.io/.

Further architectural details are in the GitHub repository: framework code is in the "App" folder, database creation scripts in the "Db" folder. Additional details on installation, technical requirements, database structure, and customization options are in the repository' s wiki pages at https://github.com/STARS4ALL/gwap-enabler/wiki.

6.2 How to Build a GWAP Using the Framework

To create a game instance from the provided template, a developer performs operations affecting the framework' s three building blocks.

First, design the basic data linking case (cf. Section 3) and atomic crowdsourcing task (cf. Section 4) for the specific use case. Then, fill the database with core resources and links (`resource`, `topic`, and `resource_{has}_{topic}` tables), GWAP aggregation parameters (`configuration` table), and badge information (`badge` table). Note that a pre-processing phase is required to prepare data, and careful analysis of the specific refinement purpose is essential for finding and tuning proper parameters. This initial step can be lengthy and complex depending on context.

Once data is in the DB, the code can run as-is or be tailored to specific requirements: game mechanics can be altered, additional resource/link description data can be added (e.g., maps/videos/sounds), or different badge/point assignment logic can be defined. Finally, customize the UI with desired look-and-feel and modify specific game elements (points, badges, leaderboards) to give the game a particular flavor or mood.

7. EXISTING APPLICATIONS BUILT ON TOP OF THE FRAMEWORK

We used the enabler to build three GWAPs addressing different data linking classes: Indomilando (link ranking), Land Cover Validation (link validation), and Night Knights (link creation and validation). Additionally, after its public open-source release, a different research group reused our framework to build their own game for collecting analogies, which we also report.

7.1 Link Ranking: Indomilando

Indomilando [26] is a web-based Game With a Purpose ranking a heterogeneous set of images depicting Milan' s cultural heritage assets. Each round shows an asset name and four photos—one representing the asset and three distractors. Users choose the correct picture and gain points for correct answers. A photo is removed after being correctly chosen three times. All answers (selection or non-selection) are recorded and analyzed ex-post to measure "how much" the picture represents the asset: the intuition is that photos correctly identified by more players are more recognizable.

In Indomilando, link set I connects each photo to its asset; assets and photos are link subjects r_s and objects r_o to be ranked. By counting and suitably weighting picture recognition instances, we calculate link scores s. Since these represent links' probability of being recognized as true, ordering them ranks the links (and thus Milan' s cultural heritage pictures) by recognizability.

Game output can serve various goals: selecting best asset pictures, understanding if an asset would benefit from additional photo collection, or evaluating if an asset needs promotional campaigns due to low recognition.

From a gamification perspective, users gain points per correct answer and can challenge others on a global leaderboard. Another incentive lets players view assets on a map with historical and cultural descriptions, as shown in Figure 3 [Figure 3: see original paper]—an additional learning reward that Indomilando players appreciated. These incentives were very effective: all 2,100 pictures were played and ranked by 151 users, achieving 125 photos ranked per hour.

7.2 Link Validation: Land Cover Validation Game

The Land Cover Validation game [38] engages Citizen Scientists in validating land cover data in Como municipality (Lombardy, Italy). Players classify areas where two land cover maps disagree: the DUSAF classification by Lombardy Region and GlobeLand30 from a Chinese agency. Validation is presented as a classification task: given an aerial photo of a 30x30 square area (pixel), players select the correct category from a predefined land use type list (e.g., residential or agricultural areas), as shown in Figure 4 [Figure 4: see original paper]. For incentives, players gain points and badges for agreeing with existing classifications and can challenge others on the leaderboard.

From a data linking perspective, each pixel is subject r_s of two links: one connecting the pixel to its DUSAF-defined land cover and another to its GlobeLand30 classification. Each time a player selects one land cover option, the corresponding link' s score s increases. This score represents link truth value, and a threshold is set so links exceeding this value are considered true. When a link score surpasses the threshold, the corresponding pixel is removed from the game.

The game fully achieved its goal: all 1,600 target aerial pixels were validated by 68 gamers playing over 20 hours during the FOSS4G Europe Conference in 2015.

7.3 Link Creation and Validation: Night Knights

Night Knights [27] is a GWAP classifying images from the International Space Station (ISS) into six fixed categories, developed within a project aiming to increase awareness about light pollution. Each participant plays an astronaut role, paired with a random player, with a mission to classify as many images as possible in one-minute rounds. As Figure 5 [Figure 5: see original paper] shows, agreeing players gain points collected to challenge others on a global leaderboard.

The hidden goal is creating new links between each image r_s and its correct category r_o by cross-validating them using multiple users' contributions, suitably aggregated. The link creation algorithm works as follows: starting from links connecting each image to all available categories, a link' s score s increases when a player chooses that category. Each image is offered to multiple players; contributions are weighted by reliability (measured via assessment tasks) and aggregated into the link score. Once the score exceeds a specific threshold, the image is classified and removed from the game. By design, a minimum of four agreeing users are required to reach the classification threshold. Since the game' s launch in February 2017, over 35,000 images have been classified by more than 1,000 players, achieving 203 images classified per hour.

7.4 Reusing and Extending the Framework: ARGO

A research group from the University of Milano Bicocca, working on analogical reasoning, contacted us seeking suggestions for designing a GWAP to involve users and collect propositional analogies (in form $a : b = c : d$). Their objectives were investigating how people create analogies and building a gold standard to train a machine learning algorithm for extracting propositional analogies from text.

After brainstorming game ideas, we introduced them to our framework. With no further support, a Master's student familiarized themselves with the software through the tutorial (see Section 9) and documentation, then customized the framework to develop ARGO, "Analogical Reasoning Game with a purpOse" – a simple game collecting analogies.

Figure 6 [Figure 6: see original paper] shows a sample screen asking users to identify which TV personality is to Italy what David Letterman is to the USA, choosing from proposed options. Here, the data linking task extends Section 3' s description: users link a partial analogy like "Letterman : USA = ?x : Italy" to the most suitable "?x" value.

During a short public event experimentation, University of Milano Bicocca researchers involved 90 users and collected around 1,200 contributions—sufficient for their further analysis and investigation.

---

8. EVALUATION OF INCREMENTAL TRUTH INFERENCE

We evaluated our truth inference algorithm through comparative assessment with alternative solutions, using partial data from two GWAPs described above: the LCV Game and Night Knights.

A first evaluation considers total contributions required. In most crowdsourcing settings with ex-post aggregation, a fixed number of contributions is collected per task. Consider multinomial classification of N tasks with L admissible labels, requiring minimum p agreeing labels per task. For ex-post aggregation with simple majority voting, total needed contributions equal redundancy r:

$$r \leftarrow N \cdot ((p-1) \cdot L + 1)$$

Moreover, traditional micro-work/crowdsourcing settings show 40%-45% spammer rates among crowd workers [39, 40], potentially increasing redundancy beyond Equation (4)' s calculation.

Table 1 shows theoretical versus empirical numbers for LCV Game and Night Knights. Our incremental approach yields significant "savings" (roughly 40%-60%) in redundancy, since when minimum contribution count p suffices to solve a task, no more labels are sought. Repetition reduction occurs because "easy" tasks are solved with minimum users (3 for LCV Game, 4 for Night Knights) due to easy agreement. Savings are higher when more alternatives exist (6 labels for Night Knights vs. 5 for LCV Game) and/or minimum user counts are higher. Conversely, savings could be smaller with high shares of "difficult" or controversial tasks.

Table 1. Number of required contributions for truth inference over N tasks.

	LCV Game	Night Knights
N	~1,000	~27,700
Theoretical	~11,500	~525,000
Actual	~6,400	~205,000

	LCV Game	Night Knights
% diff.	-44%	-61%

Note: Regarding L possible labels and minimum p agreeing answers: comparison between theoretical redundancy r (under ex-post aggregation with simple majority voting) and actual numbers measured experimentally in the two GWAPs using our incremental truth inference approach.

To assess our incremental approach' s truth inference ability, we applied state-of-the-art ex-post aggregation algorithms to contributions collected by our GWAP and compared resulting classifications. Specifically, we ran expectation maximization (EM [35]) and message passing (MP [36]), the most frequently used truth inference algorithms, then compared aggregated labels via confusion matrix. Table 2 summarizes results comparing our algorithm with EM and MP. Without ground truth, we can only compare result similarity via confusion matrix metrics. Table 2 shows very high overlap between "truths" inferred by compared algorithms (accuracy always over 96%), confirmed by agreement statistics (Kappa statistics and adjusted Rand index very high). This proves our approach' s validity and applicability.

Table 2. Truth inference results comparison between our incremental approach and state-of-the-art techniques.

Algorithms	Accuracy	Kappa	Adj. Rand
LCV Game			
incremental vs. EM	96.1%	93.4%	88.7%
incremental vs. MP	96.9%	94.7%	90.6%
Night Knights			
incremental vs. EM	99.7%	99.4%	99.4%
incremental vs. MP	99.8%	99.6%	99.6%

Note: EM: expectation maximization, MP: message passing, and various metrics: confusion matrix accuracy, Kappa statistics, adjusted Rand index corrected-for-chance [41].

9. A STEP-BY-STEP TUTORIAL TO REUSE THE FRAMEWORK

Previous sections showed how the GWAP Enabler successfully implemented games for image classification-based link creation/validation and recognizability-based link ranking. Since this is a new resource, we introduce a tutorial guiding new adopters to customize and adapt the framework to a new use case. All

required source changes are explained step-by-step in the GitHub repository at https://github.com/STARS4ALL/gwap-enabler-tutorial.

The tutorial has two goals: providing developers a guided framework "instantiation" example, and demonstrating how the GWAP Enabler can build a crowdsourcing web application to enrich and refine OpenStreetMap data (and its linked data version LinkedGeoData), motivating participation through fun incentives.

Specifically, the tutorial GWAP classifies OpenStreetMap restaurants by cuisine type. Restaurant data is selected in a specific area (Milano is given as an example, but developers can easily change location). Restaurants with already-specified cuisine types serve as "ground truth" (for assessment tasks computing player reliability), while remaining restaurants are subject resources targeted for classification.

Players are randomly paired and shown restaurant names and positions on a dynamic map; the game involves agreeing on cuisine type selected from predefined categories (most widely used in OpenStreetMap). Player contributions are aggregated through truth inference implementing link collection and validation.

Creating this application requires framework core functionality changes. While the default GWAP Enabler displays resources as images, this tutorial scenario shows developers how to display both textual information (restaurant name) and an interactive map (restaurant position). This requires: (1) correctly storing relevant information in the database, (2) modifying API code to retrieve additional data, and (3) modifying UI code to display name and map.

The tutorial provides basic instructions and explains map embedding using Leaflet, an open-source JavaScript library for mobile-friendly interactive maps. We don' t detail graphical aspects, letting developers define desired look-and-feel to give the game personal flavor. Following wiki instructions, developers can have a working GWAP in about half a day and gain sufficient framework knowledge for reuse, as the tutorial covers all relevant modifications.

University of Milano Bicocca people used this tutorial to create the ARGO game described in Section 7.4. They reported that in less than a week, they realized their vision by designing the experiment, preparing data, implementing the game on our framework, and deploying it. This experience testifies that our GWAP Enabler is indeed an easy-to-use and easy-to-extend resource.

---

10. DISCUSSION AND CONCLUSION

This paper presented an open-source software framework for building Games with a Purpose embedding crowdsourcing tasks for linked data refinement. The framework helps collect manually-labeled gold standard data needed as training sets for automatic learning algorithms implementing refinement operations (link

collection, ranking, validation) on large-scale linked datasets and knowledge graphs. In other words, the framework simplifies the tedious and expensive human data collection process, letting researchers focus on subsequent scientific study and experimentation steps.

We introduced data linking cases implemented by the framework to explain its generality and reuse potential; illustrated the crowdsourcing task and truth inference process to clarify design and customization possibilities; provided technical details about the GWAP Enabler internals, designed and developed according to common web development best practices; and demonstrated framework versatility through diverse applications and experimental evaluation of our incremental truth inference approach.

Since our framework not only collects data linking task solutions from game players but also automatically computes data aggregation via a specific truth inference approach, we introduced our incremental algorithm, explaining its rationale, implementation, and providing qualitative and quantitative evaluation of its benefits.

Finally, we presented a step-by-step tutorial as detailed documentation to ease reuse of this new resource. Following the tutorial, a developer is guided to build an entirely new GWAP in a few hours, saving significant coding effort.

We provided all references to access framework code (released under Apache 2.0 license) and online documentation including data schemas, API specifications, sample input/output data, technical requirements, installation instructions, and guided customization instructions.

The framework could be further extended to cover other refinement cases like free-text labeling (i.e., insertion of new literal objects) or data linking issues related to choosing different predicates.

AUTHOR CONTRIBUTIONS

The ideas and concepts presented result from at least three years of cooperation between the authors. I. Celino (irene.celino@cefriel.com) focused on data linking, crowdsourcing tasks, and incremental truth inference; G. Re Calegari (gloria.re@cefriel.com) and A. Fiano (andrea.fiano@cefriel.com) focused on the framework, its applications, evaluation, and tutorial. All authors contributed to manuscript writing and edited/reviewed the final article.

ACKNOWLEDGEMENTS

This work was partially supported by the STARS4ALL project (H2020-688135) co-funded by the European Commission.

REFERENCES

[1] H. Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web* 8(3)(2017), 489–508. doi: 10.3233/SW-160218.

[2] A.J. Quinn & B.B. Bederson. Human computation: A survey and taxonomy of a growing field. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 1403–1412. doi: 10.1145/1978942.1979148.

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

[4] I. Celino, A. Fiano & R. Fino. Analysis of a cultural heritage game with a purpose with an educational incentive. In: *International Conference on Web Engineering*, 2016, pp. 422–430. doi: 10.1007/978-3-319-38791-8_{28}.

[5] L. Von Ahn & L. Dabbish. Designing games with a purpose. *Communications of the ACM* 51(8)(2008), 58–67. doi: 10.1145/1378704.1378719.

[6] I.P. Fellegi & A.B. Sunter. A theory for record linkage. *Journal of the American Statistical Association* 64(328)(1969), 1183–1210. doi: 10.2307/2286061.

[7] A. Ferrara, A. Nikolov & F. Scharffe. Data linking for the semantic web. In: *Semantic Web: Ontology and Knowledge Base Enabled Tools, Services, and Applications*, 2013, 169–200. doi: 10.4018/978-1-4666-3610-1.ch008.

[8] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann & S. Auer. Quality assessment for linked data: A survey. *Semantic Web* 7(1)(2016), 63–93. doi: 10.3233/SW-150175.

[9] C. Fürber & M. Hepp. Using SPARQL and SPIN for data quality management on the semantic web. In: W. Abramowicz & R. Tolksdorf (eds.) *Business Information Systems (BIS 2010)*. Berlin: Springer, pp. 35–46. doi: 10.1007/978-3-642-12814-1_4.

[10] C. Guéret, P. Groth, C. Stadler, & J. Lehmann. Assessing linked data mappings using network measures. In: *Extended Semantic Web Conference*, 2012, pp. 87–102.

[11] H. Paulheim & C. Bizer. Improving the quality of linked data using statistical distributions. *International Journal on Semantic Web and Information Systems (IJSWIS)* 10(2)(2014), 63–86.

[12] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy ···& W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 601–610. doi: 10.1145/2623330.2623623.

[13] J. Sleeman & T. Finin. Type prediction for efficient coreference resolution in heterogeneous semantic graphs. In: *Semantic Computing (ICSC), 2013 IEEE Seventh International Conference*, 2013, pp. 78–85. doi: 10.1109/ICSC.2013.22.

[14] J. Sleeman, T. Finin & A. Joshi. Topic modeling for RDF graphs. In: *Workshop on Linked Data for Information Extraction (LD4IE) at ISWC*, 2015, pp. 48-62.

[15] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer & J. Lehmann. Crowdsourcing linked data quality assessment. In: *International Semantic Web Conference*, 2013, pp. 260–276. doi: 10.1007/978-3-642-41338-4_{17}.

[16] E. Simperl, B. Norton & D. Vrandecic. Crowdsourcing tasks in linked data management. In: *Proceedings of the Second International Conference on Consuming Linked Data*, 2012, pp. 61–72.

[17] K. Siorpaes & M. Hepp. Games with a purpose for the semantic web. *IEEE Intelligent Systems* 23(3)(2008). doi: 10.1109/mis.2008.45.

[18] J. Waitelonis, N. Ludwig, M. Knuth & H. Sack. WhoKnows? Evaluating linked data heuristics with a quiz that cleans up DBpedia. *Interactive Technology and Smart Education* 8(4)(2011), 236–248. doi: 10.1108/17415651111189486.

[19] J. Chamberlain, M. Poesio & U. Kruschwitz. Phrase detectives: A web-based collaborative annotation game. In: *Proceedings of the International Conference on Semantic Systems (I-Semantics' 08)*, 2008, pp. 42–49.

[20] S. Thaler, E.P.B. Simperl & K. Siorpaes. SpotTheLink: A game for ontology alignment. *Wissensmanagement* 182(2011), 246–253.

[21] J. Hees, T. Roth-Berghofer, R. Biedert, B. Adrian, & A. Dengel. BetterRelations: Using a game to rate linked data triples. In: *Annual Conference on Artificial Intelligence*, 2011, pp. 134–138.

[22] I. Celino, S. Contessa, M. Corubolo, D. Dell' Aglio, E. Della Valle, S. Fumeo & T. Krüger. Linking smart cities datasets with human computation: The case of UrbanMatch. In: P. Cudré-Mauroux et al. (eds.) *The Semantic Web -ISWC 2012*. Berlin: Springer, pp. 34–49. doi: 10.1007/978-3-642-35173-0_3.

[23] I. Celino, D. Cerizza, S. Contessa, M. Corubolo, D. Dell' Aglio, E.D. Valle, & S. Fumeo. Urbanopoly: A social and location-based game with a purpose to crowdsource your urban data. In: *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom)*, 2012, pp. 910–913. doi: 10.1109/SocialComPASSAT.2012.138.

[24] C. Wieser, F. Bry, A. Bérard & R. Lagrange. ARTigo: Building an artwork search engine with games and higher-order latent semantic analysis. In: *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[25] I. Celino. On the effectiveness of a mobile puzzle game UI to crowdsource linked data management tasks. In: *1st International Workshop on User Interfaces for Crowdsourcing and Human Computation*, 2014.

[26] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6(1)(2012), 1–114.

[27] G. Re Calegari, G. Nasi & I. Celino. Human computation vs. machine learning: An experimental comparison for image classification. *Human Computation Journal* 5(1)(2018), 13–30.

[28] L. Von Ahn & L. Dabbish. Labelling images with a computer game. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2004, pp. 319–326. doi: 10.1145/985692.985733.

[29] U. Ul Hassan, S. O' Riain & E. Curry. Effects of expertise assessment on the quality of task routing in human computation. In: *Proceedings of the 2nd International Workshop on Social Media for Crowdsourcing and Human Computation*, 2013. doi: 10.14236/ewic/sohuman2013.1.

[30] Y. Zheng, G. Li, Y. Li, C. Shan & R. Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment* 10(5)(2017), 541–552. doi: 10.14778/3055540.3055547.

[31] V.S. Sheng, F. Provost & P.G. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 614–622.

[32] B. Mozafari, P. Sarkar, M.J. Franklin, M.I. Jordan & S. Madden. Active learning for crowd-sourced databases. arXiv preprint. arXiv:1209.3686, 2012.

[33] B. Mozafari, P. Sarkar, M. Franklin, M. Jordan & S. Madden. Scaling up crowd-sourcing to very large datasets: A case for active learning. In: *Proceedings of the VLDB Endowment* 8(2)(2014), 125–136.

[34] E. Law & L.v. Ahn. Human computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 5(3)(2011), 1–121. doi: 10.2200/S00371ED1V01Y201107AIM013.

[35] A.P. Dawid & A.M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics*(1979), pp. 20–28. doi: 10.2307/2346806.

[36] D.R. Karger, S. Oh & D. Shah. Iterative learning for reliable crowdsourcing systems. In: *Advances in neural information processing systems*, 2011, pp. 1953–1961.

[37] D.C. Brabham. *Crowdsourcing*. Cambridge, MA: MIT Press, 2013.

[38] M.A. Brovelli, I. Celino, A. Fiano, M.E. Molinari & V. Venkatachalam. A crowdsourcing-based game for land cover validation. *Applied Geomatics*

10(1)(2018), 1–11. doi: 10.1007/s12518-017-0201-3.

[39] R. Shah. Spam hurts crowdsourcing but can' t kill it (Forbes Contributor Opinions). Available at: https://www.forbes.com/sites/rawnshah/2010/12/17/spam-hurts-crowdsourcing-but-cant-kill-it.

[40] J. Vuurens, A.P. de Vries & C. Eickho. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In: *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR' 11)*, 2011, pp. 21–26.

[41] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* 66(336)(1071), 846–850. doi: 10.2307/2284239.

AUTHOR BIOGRAPHY

Irene Celino is Head of the Knowledge Technologies group at Cefriel, where she leads an R&D team and serves as Portfolio and Project Manager. With expertise in Semantic Web and Human Computation technologies, her research applies these innovations to design and develop web applications, search engines, recommendation systems, and mobile games, particularly in Smart City and transportation scenarios. She has over 15 years of experience in 30+ R&D cooperative projects at National/Regional and European levels (FP6, FP7, H2020, EIT Digital) and has authored 70+ scientific publications in peer-reviewed journals, books, and conferences. ORCID: 0000-0001-9962-7193

Gloria Re Calegari is a researcher at Cefriel with a computer science background. Her expertise lies in Data Science and Human Computation technologies, covering design and development of gamified applications and Games with a Purpose, alongside machine learning solutions that integrate humans and artificial intelligence. During her 5+ years in R&D cooperative projects at National and Regional levels, she has published 20+ scientific papers in peer-reviewed journals and conferences. ORCID: 0000-0002-4558-229X

Andrea Fiano is a senior developer at Cefriel. Starting with .Net web application development, he progressed to Java backend solutions and REST APIs, and gained experience with Single Page Applications and Progressive Web Apps using Angular and Node.js. He provides expertise in developing customer-tailored solutions and supporting research, particularly in Human Computation through development of Games with a Purpose for Smart City and Crowdsourcing scenarios. ORCID: 0000-0002-0963-4689

Note: Figure translations are in progress. See original paper for figures.