Crowd-assessing quality in uncertain data linking datasets

Article *in* The Knowledge Engineering Review · July 2020

DOI: 10.1017/S0269888920000363

5 authors, including:

Daniel Faria
Inesc-ID
56 PUBLICATIONS **2,276** CITATIONS
SEE PROFILE

Alfio Ferrara
University of Milan
116 PUBLICATIONS **1,926** CITATIONS
SEE PROFILE

Ernesto Jiménez-Ruiz
City, University of London
154 PUBLICATIONS **2,610** CITATIONS
SEE PROFILE

Stefano Montanelli
University of Milan
102 PUBLICATIONS **1,299** CITATIONS
SEE PROFILE

Crowd-assessing Quality in Uncertain Data Linking Datasets

Daniel Faria[1,2], Alfio Ferrara[3,4], Ernesto Jiménez-Ruiz[5,6], Stefano Montanelli[3,4], and Catia Pesquita[7]

[1] *Instituto Gulbenkian de Ciência, Oeiras, Portugal*
E-mail: dfaria@igc.gulbenkian.pt
[2] *INESC-ID, Lisboa, Portugal*
[3] *Department of Computer Science, Università degli Studi di Milano, Milan, Italy*
E-mail: {alfio.ferrara,stefano.montanelli}@unimi.it
[4] *Data Science Research Center, Università degli Studi di Milano, Milan, Italy*
[5] *City, University of London, UK*
E-mail: ernesto.jimenez-ruiz@city.ac.uk
[6] *Department of Informatics, University of Oslo, Norway*
E-mail: ernestoj@ifi.uio.no
[7] *Lasige, Faculdade de Ciências, Universidade de Lisboa, Portugal*
E-mail: clpesquita@fc.ul.pt

Abstract

The quality of a dataset used for evaluating data linking methods, techniques, and tools depends on the availability of a set of mappings, called *reference alignment*, that is known to be correct. In particular, it is crucial that mappings effectively represent relations between pairs of entities that are indeed similar due to the fact that they denote the same object. Since the reliability of mappings is decisive in order to perform a fair evaluation of automatic linking methods an tools, we call this property of mappings as *mapping fairness*. In this article, we propose a crowd-based approach, called Crowd Quality (CQ), for assessing the quality of data linking datasets by measuring the fairness of the mappings in the reference alignment. Moreover, we present a real experiment, where we evaluate two state of the art data linking tools before and after the refinement of the reference alignment based on the CQ approach, in order to present the benefits deriving from the crowd-assessment of mapping fairness.

1 Introduction

Data linking is the activity of joining different sources of data by interrelating the datum therein. This involves automatically determining references to the same objects and relations between related objects, usually by comparing their data descriptions in the different sources. The Semantic Web has been one of the main research areas where this problem has been studied.

Ontology matching is the activity of discovering relations, called mappings, between entities in an ontology (Euzenat et al., 2007). Often, the term is reserved to the the matching of concepts and properties in the schema (or Tbox) of an ontological description, whereas instance matching is used to refer to the matching of individuals described at the instance level (or Abox). Data linking of linked data relies heavily on instance matching, and the two terms are often used interchangeably. In particular, we focus our work mainly on instance matching deduplication tasks, where links between entities are represented by owl:sameAs relations.[1]

Several approaches and tools for ontology matching have been proposed along the last 20 years (Euzenat and Shvaiko, 2013; Algergawy et al., 2018), and the field received significant

[1] Throughout this article, we use *ontology matching* or simply *matching* to denote the matching of either Tbox concepts/properties or the matching of Abox individuals, and *ontology* to denote either the Tbox or the Abox. We use *instance matching* when referring specifically to the matching of Abox individuals.

attention since 2004 thanks also to the Ontology Alignment Evaluation Initiative (OAEI)[2]. In order to compare tools, appropriate methods and techniques are needed to evaluate their performance in terms of their capability of effectively detecting the correct mappings between two ontology descriptions and/or two ontology instances.

The conventional approach to evaluation is based on the comparison of the mappings automatically retrieved by a tool against a set of correct mappings, called reference alignment, which is manually defined by experts of the domain at hand in order to ensure a ground truth for the evaluation. However, the increasing size of the data to be compared as well as the lack of a gold standard, especially when dealing with instances, makes it difficult and time consuming to manually define such a reference alignment. For this reason, it is becoming quite common to rely on methods and tools for *synthetic dataset generation* where the reference alignment is obtained by systematically applying a set of transformations to an initial dataset (source), in order to create a transformed set of data to be matched against the initial one (target). The idea is that tools will have to discover correspondences (mappings) between source and target entities that have been derived through transformations (Ferrara et al., 2011; Euzenat et al., 2013; Saveta et al., 2015; Röder et al., 2017). On one hand, these methods have the advantage of being scalable with respect to the number of entities in the source dataset and they usually provide parameters that can be used to generate controlled transformations in order to design the reference alignment in a disciplined manner. On the other hand, it is quite difficult to set up the correct values of the configuration parameters in order to achieve realistic and fair transformations. Moreover, it is difficult to understand how much a specific transformation will impact the performance of matching tools and if these tools' results will be affected by a transformation the same way as the human judgment would be.

In the context of ontology matching evaluation, the notion of mapping correctness is related to the human judgment about the similarity between the entities. Thus, it is clear that, if the goal of data transformation approaches is to provide a fair evaluation of matching tools in alternative to manually curated reference alignments, then the crucial issue is to determine whether the transformations result in mappings that would be considered correct by human standards. If the transformations lead to unrecognizable property values to such an extent that it is no longer plausible for a human to consider that the entities with those values denote the same object, then matching tools should not be penalized for missing them. On the contrary, they should be rewarded for it, as finding such extreme transformations could very well lead to finding false positives in real matching scenarios. As such, transformations that are too extreme do not provide a fair benchmark for evaluating ontology matching tools.

In this paper, we introduce the notion of *mapping fairness* to denote how much a mapping generated through the application of data transformations can be considered a "plausible mapping" to be recognized by a matching tool. To this end, we propose a crowd-based approach, called Crowd Quality (CQ) to determine the degree of fairness of a mapping by relying on the human judgement. The ultimate goal is to assess the quality of a data linking dataset by measuring the fairness of the mappings in the reference alignment. The idea is that the fairness of the evaluation of ontology matching tools depends on the number of fair mappings that are available in the reference alignment (Carmines and Zeller, 1979). In particular, if a mapping is fair, it is also fair to require a matching tool to discover it. On the opposite, it is unfair to expect a matching tool to discover a mapping which is itself unfair. Borrowing the idea presented in Cheatham and Hitzler (2014), we argue that crowdsourcing techniques can be successfully employed for evaluating the degree of mapping fairness in a given reference alignment, with a specific focus on alignments automatically generated by transformations. In particular, we agree with Cheatham and Hitzler (2014) that the human interpretation still remains the crucial capability to capture the meaning of a mapping and to properly rate the quality of a link between two similar items. On this point, we present crowdsourcing techniques characterized by the combined use of *range*

[2] `http://oaei.ontologymatching.org`.

tasks and *consensus mechanisms* for enforcing a fine-evaluation of mapping fairness that really express the human-perceived judgement. On top of the fairness evaluation, we aim at supporting new evaluation measures, beyond those based on Precision and Recall (see Section 2), capable of taking into account the mapping fairness in the evaluation of ontology and instance matching tools in order to make the evaluation more fair. Our work is focused on instance matching but the CQ approach can be easily extended to cope with schema-level ontology matching.

The paper is organized as follows. In Section 2, we discuss our motivations and we provide the main background definitions used in this work. In Section 3 and 4, we present the CQ approach and the techniques exploited for the evaluation of mappings using crowdsourcing, respectively. Lessons learned in a real experiment using two state of the art instance matching tools and the crowd are discussed in Section 5. In Section 6, we provide an overview of the recent literature on data linking evaluation and crowdsourcing techniques for mapping validation. Finally, in Section 7, we give our concluding remarks.

2 Motivations and background

A dataset for the evaluation of ontology matching tools is typically characterized by a *source ontology* O, a *target ontology* O', and a *reference alignment* $\mathcal{E}$ that is a set of mappings $m(i, j)$ between an entity $i \in O$ and an entity $j \in O'$. During the evaluation, an ontology matching tool is used to systematically compare entities of O against entities of O' with the aim to retrieve an alignment $\mathcal{R}$. In this context, the reference alignment $\mathcal{E}$ contains the mappings that are *expected* to be retrieved and it serves as *ground truth* to exploit for quality evaluation of the alignment $\mathcal{R}$. In particular, given $\mathcal{E}$ and $\mathcal{R}$, each mapping $m(i, j)$ is classified as in Table 1. When a mapping

	$m(i,j) \in \mathcal{R}$	$m(i,j) \notin \mathcal{R}$
$m(i,j) \in \mathcal{E}$	TP	FN
$m(i,j) \notin \mathcal{E}$	FP	TN

Table 1 Classification of mappings with respect to $\mathcal{R}$ and $\mathcal{E}$

is retrieved and expected, it is said to be a *true positive* (TP) result. *False negatives* (FN) are mappings that are expected but not retrieved, while *false positive* (FP) are mappings retrieved but not expected. Finally, *true negatives* (TN) are mappings that were not expected and have not been retrieved. Of course, a perfect ontology matching tool should retrieve what is expected (TP) and should not retrieve what is not expected (TN). However, since the set of *true negatives* is usually large and completely well-defined by the other options (i.e., $(O \times O') - \mathsf{TN} \equiv \mathsf{TP} \cup \mathsf{FP} \cup \mathsf{FN} \equiv \mathcal{R} \cup \mathcal{E}$), the evaluation of ontology matching tools does not use TN and it is typically based on the measures of *Precision* and *Recall*, defined as follows:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}$$

The classical Precision and Recall are conceived to work with boolean mappings, which state that a correspondence between two entities i and j can be just true or false; however, this is not how the majority of ontology matching tools work. The result is usually a mapping of the form $m(i, j, \sigma)$, where $\sigma \in [0, 1]$ is a measure of similarity between i and j. When evaluating this kind of mappings, it is common to exploit the following two strategies. On one side, a threshold over σ is used to decide which mappings should be included in the retrieved results. On the other side, the measure σ is used to define a ranking of the retrieved results, from the most similar mappings to the less similar ones. In such a way, we can observe the Precision of the tool at different levels of Recall. In particular, when we scan the ranking from the top to the bottom, the Recall increases and we can measure the corresponding degree of Precision by calculating the *Interpolated Precision* $I^p(r)$ at the level r of recall as $I^p(r) = \max_{r' \geq r} p(r')$, that is the highest Precision found for any recall $r' \geq r$. This technique provides an evaluation which takes into

account the level of similarity calculated for each mapping by the matching tools and makes it possible to describe the performance of each tool by studying how the Precision changes with respect to Recall.

2.1 Mapping fairness

Both classical Precision/Recall evaluation and ranked evaluation share the idea that all the mappings in the reference alignment $\mathcal{E}$ are equally *fair*. In this paper, we propose an approach called CQ to annotate each mapping of the reference alignment with a measure of its fairness, that is defined as follows.

Definition 2.1 Mapping fairness. *Given a mapping $m(i, j)$, its fairness $\rho(m(i, j)) \rightarrow [0, 1]$ is a measure that estimates how supported the mapping is by the information in the ontologies, from a human perspective. A fairness of 0 means that no human could plausibly agree that the mapping is correct, given the available information, whereas a fairness of 1 means that no human would doubt the correctness of the mapping.*

While the concept of mapping fairness can be applied to traditional reference alignments produced by human experts, as in the work of Cheatham and Hitzler (2014), it is especially relevant for synthetically generated datasets, where the reference alignment is defined by applying a set of transformations to entities of the source ontology in order to obtain their matching counterpart in the target ontology. The following is the general definition of *transformation*.

Definition 2.2 Transformation. *Let s be a source entity and S the set of ontology axioms/assertions of the form $\langle s, p, o\rangle$, where s is the subject, p denotes a property and o is the object. A transformation $\tau(S, p, w) \rightarrow T$ is a function that maps s on a target counterpart s' (i.e., $m(s, s')$) and transforms S in a new set T of axioms/assertions by applying a specific transformation with strength w to each assertion/axiom having p as a property.*

Example. Consider the following two transformations[3]. Let game001 be an individual in the source ontology that describes a board game. The set S of source ontology assertions describing game001 is the following:

S (source ontology assertions about game001)		
game001	rdf:type	Boardgame
game001	dc:title	"Civilization: the new world"
game001	dc:date	2012

In our example, we define two transformations, namely char_mod and property_del. Given a property p and a transformation strength w, the char_mod transformation changes the string literal object o of each assertion on the property p by randomly substituting a number of characters equal to $\lceil w \cdot len(o) \rceil$. The property_del transformation instead deletes each assertion on p with probability w. Now, we generate two different matching counterparts for game001, with two different corresponding reference mappings, namely m(game001, game002) and m(game001, game003). The two mappings m(game001, game002) and m(game001, game003) are generated as follows:

$$m(\text{game001}, \text{game002}) \leftarrow \text{char\_mod}(S, \text{dc:title}, 0.1)$$
$$m(\text{game001}, \text{game003}) \leftarrow \text{property\_del}(\text{char\_mod}(S, \text{dc:title}, 0.8), \text{dc:date}, 1.0)$$

In m(game001, game002) we just modify the game title by randomly substituting 2 characters. Instead, m(game001, game003) is generated by first changing 21 characters in the game title and then, on top of the char_mod transformation, by deleting the date with probability 1.0. The two resulting individual descriptions T' and T'' are the following:

[3]The running example is based on an ontology Abox describing about 1 600 Boardgames. Data have been retrieved from the BoardGameGeek (BGG) website (`https://boardgamegeek.com`).

T' (target ontology assertions about game002)

game002	rdf:type	Boardgame
game002	dc:title	"CivilizaDion: ghe new world"
game002	dc:date	2012

T'' (target ontology assertions about game003)

game003	rdf:type	Boardgame
game003	dc:title	"bbvolizatioQ7 tkVuMewIuorl5"

The example shows how a different application of transformations may produce mappings with a different fairness. In the first case (i.e., $m(\mathsf{game001}, \mathsf{game002})$) retrieving the mapping means to compare two entity descriptions which share the same type and date (i.e., Boardgame and 2012) and a title that is only slightly different from the original. Thus, $m(\mathsf{game001}, \mathsf{game002})$ can be considered a highly reliable mapping which is "fair" to expect in the results of an ontology matching tool. On the contrary, in $m(\mathsf{game001}, \mathsf{game003})$, the two entity descriptions share only generic information (i.e., the type) but not a date nor a title in that the title has been heavily changed and it is remarkably different from the original one. Thus, this last mapping is not reliable and it is "unfair" to pretend that a matching tool will be able to retrieve it.

The impact of a transformation on the target ontology and, as a consequence, on the difficulty of the matching tasks may be measured by introducing the notion of *quantity of transformation*. Measuring how much the source ontology is changed during the transformation process is important in this context because the number of data values, properties, and logical values that are changed has a direct influence on how much it is potentially difficult for a matching tool to find a mapping between the source entity and its transformed counterpart. The quantity of transformation is defined as follows.

Definition 2.3 Quantity of transformation. *The quantity of transformation T_i^Q affecting a source ontology entity i is a vector where each value is a score associated with a feature describing the process of mapping production. In case of transformations, a feature can represent each individual transformation or the set of transformations grouped by property or by axiom/assertion. The scores are intended to represent how easy it will be for a matching tool to detect the correspondence at the property or the axiom/assertion value, with high scores denoting easy matching tasks and low scores denoting difficult matching tasks.*

T_i^Q scores may be set up manually, by enforcing a heuristic approach that associates a given score to each kind of transformation or that just evaluates the relevance of each property in the matching process in case of manually defined mappings. As an alternative, scores may be defined automatically according to the strength w associated with each transformation.

Example. Assume to have an individual s in the source ontology and three data property assertions of the form $\langle s, p^1, o^1\rangle$, $\langle s, p^1, o^2\rangle$, and $\langle s, p^2, o^3\rangle$. If the quantity of transformation is assessed by grouping transformations by property, we will have a vector T_s^Q over two dimensions, each one reporting the score associated with the transformations applied to p^1 and p^2, respectively. Thus, in case of a transformation which involves both the first and the second assertion, the score $T_s^Q[0]$ represents the cumulative effect of applying the transformation to both the assertions involving the property p^1.

3 Proposed approach

The CQ approach (see Figure 1) is based on the use of crowdsourcing techniques for fairness assessment of a given ontology alignment, which can either be a manually-defined alignment or a synthetic alignment created by a set of transformations operated on the source ontology (Figure 1 (A)).

The first step in the CQ approach is to assess the quantity of transformation T_i^Q associated with each mapping m_i in the reference alignment $\mathcal{E}$ (Figure 1 (B)). In the second step of CQ

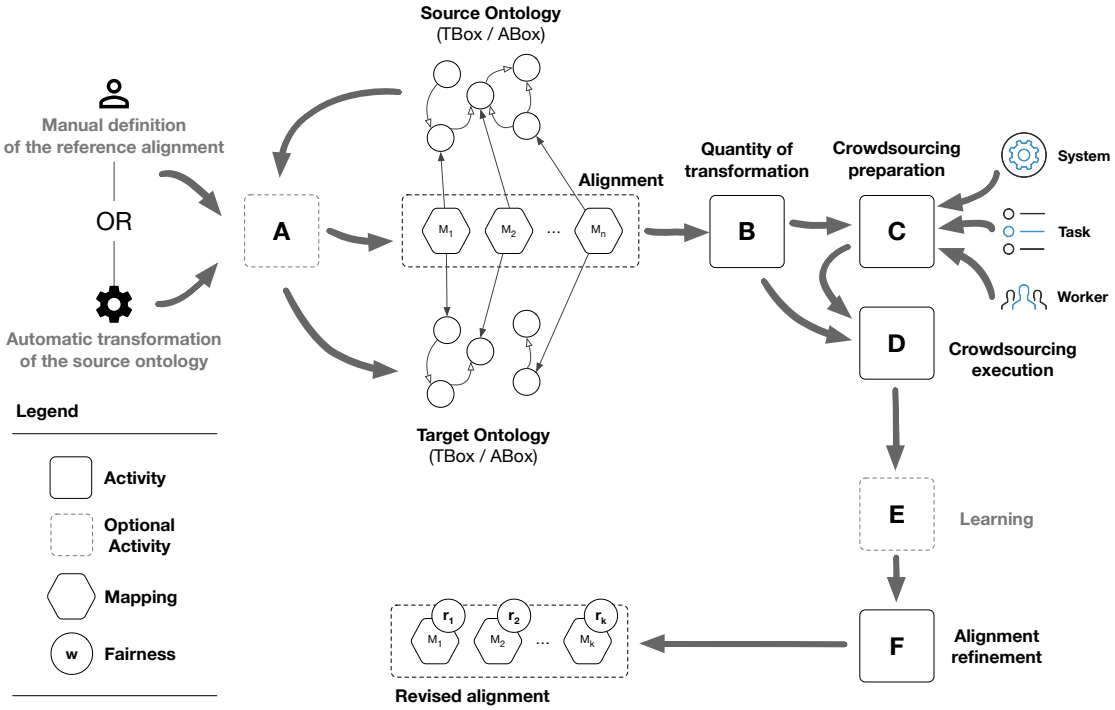

Figure 1 The CQ approach for fairness assessment of ontology alignment

(Figure 1 (C)), we configure the crowdsourcing framework for supporting the fairness evaluation of the mappings in $\mathcal{E}$. In particular, crowd workers are involved in the execution of a set of tasks $\mathcal{T} = \mathcal{T}^{\mathcal{E}} \cup \mathcal{T}^{\mathcal{F}}$ composed by tasks $\mathcal{T}^{\mathcal{E}}$ taken from the reference alignment $\mathcal{E}$, and tasks $\mathcal{T}^{\mathcal{F}}$ that are generated as a gold standard *control set* to check the quality of the crowd work (see Section 4). The tasks $\mathcal{T}$ are submitted to the crowd for evaluation. On one side, we expect that the workers assign high degrees of fairness to the tasks $\mathcal{T}^{\mathcal{E}} \subseteq \mathcal{T}$ associated with correct mappings, thus confirming that the descriptions i and j of a mapping $m(i, j)$ actually refer to the same real entity. On the other side, we expect that the crowd workers associate low degrees of fairness to the mapping $\mathcal{T}^{\mathcal{F}} \subseteq \mathcal{T}$. Independently from the quality of the transformation process, we expect such behavior of crowd workers since a fake mapping for the control set is established between a pair of different, separate entity descriptions. In order to make fake mappings more challenging so that they cannot be trivially evaluated as unfair, we map entities in the source ontology with other transformed entities in the target ontology by exploiting the following criterion: 1) given an entity i in the source ontology, we take the set S of all the entities in the source ontology that are either siblings of i or instances of the same classes of i, according to the fact that i is a class or an individual, respectively. Then, we randomly pick up a transformed entity j' from the set of all the transformations derived from entities in S and we create a task for the mapping $m(i, j')$. An example of this procedure in shown in Figure 2.

In the example, a transformation applied to three individuals (i.e., I_1, I_2, I_3) in a source Abox produces three transformed individuals (i.e., $I_1^{'}$, $I_2^{'}$, $I_3^{'}$) and the corresponding set of mappings M_1, M_2, and M_3 in the reference alignment $\mathcal{E}$. In order to create a gold standard for crowd evaluation, we need to define not only a task for each of the (true) mappings but also the control set $\mathcal{T}^{\mathcal{F}}$ based on "fake mappings", that are mappings between unrelated individuals (i.e., the target individual was not produced by transforming the source individual). To ensure that fake mappings are plausible and not trivial to detect as incorrect, we only generate such mappings between instances of the same class. Instances of different classes are likely to have very different properties in their axioms and thus be trivial to identify. Thus, to generate fake mappings, we

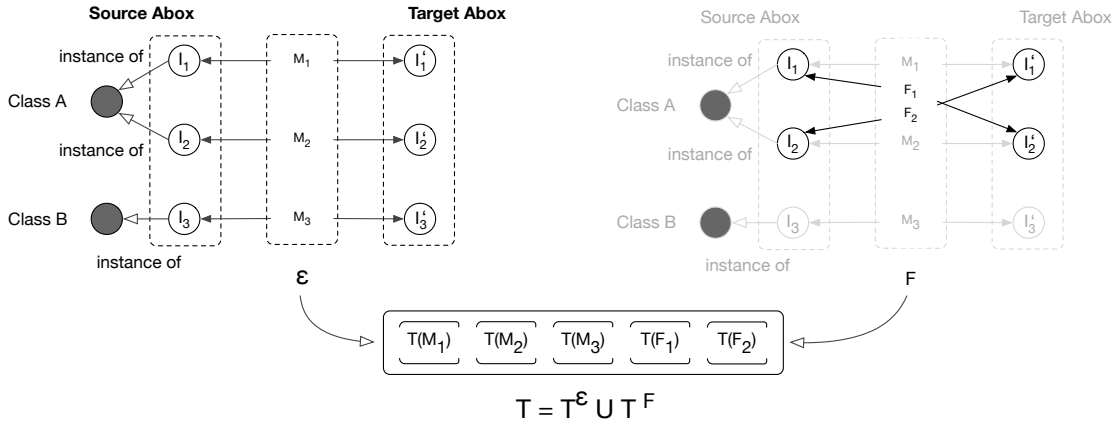

Figure 2 Example of the building process of the tasks $\mathcal{T}$

randomly select from the gold standard the target individual to be mapped on I_1 among those that are matching counterparts of the individuals that are instance of the same class than I_1. In the example, this means that we select $I_2^{'}$, because we have $m(I_2, I_2^{'})$, $A(I_2)$ and $A(I_1)$. By applying this criterion to I_1 and I_2, we produce two new mappings $F_1 = m(I_1, I_2^{'})$ and $F_2 = m(I_2, I_1^{'})$. Finally, for each mapping M_i either in the reference alignment or in the set of fake mappings, we define a task $T(M_i)$ in the final task set $\mathcal{T} = \mathcal{T}^{\mathcal{E}} \cup \mathcal{T}^{\mathcal{F}}$.

The third step in CQ (Figure 1 (D)) is based on the crowdsourcing execution of the tasks $\mathcal{T}$. It is important to consider that a basic principle of crowdsourcing is about rewarding workers for their effort in task execution. For this reason, the number of tasks $\mathcal{T}$ submitted to the crowd can be different from the number of mappings in the reference alignment $\mathcal{E}$ due to budget constraints. Moreover, each task is assigned to a work force composed of multiple independent workers, thus it is possible that a final result is not collected for all the tasks $\mathcal{T}$ due to insufficient crowd participation. As a result, it may happen that a crowd-based fairness evaluation is available only for a subset of the mappings in the reference alignment $\overline{\mathcal{E}} \subseteq \mathcal{E}$. In this case, a learning step is enforced (Figure 1 (E)) in order to predict the crowd fairness evaluation for the missing mappings. This learning step relies on the features and the quantity of transformation T^Q that has been applied to a mapping task for which a crowd result has been collected. In particular, we use the subset of tasks that have been executed by the crowd as a training set for regression with the goal of predicting the fairness values given the mappings and the quantity of transformation.

Finally, the refinement of the reference alignment $\mathcal{E}$ is performed in CQ (Figure 1 (F)). The refinement step is first based on annotating each mapping $m_i \in \mathcal{E}$ with the fairness degree $\rho(m_i)$ provided by the crowd. This new alignment $\mathcal{E}^R$ provides a representation of the mapping fairness in a continuous space. In order to transform fairness in a categorical judgment over the mappings validity, we specify a threshold th over the crowd fairness degrees[4] such that the refined alignment is defined as $\mathcal{E}' = \mathcal{E} - \{m_i \mid \rho(m_i) < th\}$. Given the annotated reference alignment $\mathcal{E}^R$ and the fairness degrees F^R provided by the crowd for the fake tasks $\mathcal{T}^{\mathcal{F}}$, we define the following crowd-error function $C_e(r) \rightarrow [0, 1]$.

$$C_e(r) = \frac{\mid \{m_i \in \mathcal{E}^R : \rho(m_i) < r\} \mid + \mid \{f_i \in F^R : \rho(f_i) > r\} \mid}{\mid \mathcal{E}^R \mid + \mid F^R \mid}, \tag{1}$$

where r denotes a fairness degree value. The value of $C_e(r)$ is inversely proportional to the number of mappings with high fairness values that are correct (i.e., included in the reference alignment)

[4]The setting of the refinement threshold th is a parameter that is expected to be set up by the designer of the evaluation process. The idea is that higher values of th produce a simpler challenge for matching tools in that only highly fair mappings are preserved. In this paper, we run an extensive experimental evaluation of the impact of different levels of th on the evaluation process (see Section 5).

and with the number of tasks with low fairness values that are actually fake, which states the error of the crowd in its fairness judgment. The threshold for determining the class of fair mappings is thus the value of r which minimizes $C_e(r)$.

4 Crowdsourcing techniques

The term crowdsourcing has been firstly introduced in Howe (2006) and it is defined as a type of participatory online activity in which an individual or an institution proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task (Arolas and de Guevara, 2012).

In CQ, crowdsourcing techniques are employed for human-assessment of mapping fairness. Given a mapping, a corresponding crowdsourcing task is assigned to multiple crowd-workers for evaluation. This kind of task is usually known as *collective task* and the final task result (i.e., the evaluation of mapping fairness) depends on the personal knowledge, perception, and expertise of the human workers involved in task execution. Since many different answers are collected from crowd workers on a given task, a consensus mechanism is enforced to determine the final task evaluation/result on the basis of the different obtained opinions. In other words, CQ relies on the so-called "wisdom of the crowd", in which the fairness of a task result (i.e., the fairness of a mapping) is determined by its credibility: the higher the consensus among workers on an answer, the more fair the task is (Castano et al., 2016).

Conventional crowdsourcing solutions for execution of collective tasks are based on choice questions where the possible task answers are predefined and the worker executes the task by choosing the preferred option among those available (Sarasua et al., 2012; Cruz et al., 2014; Cheatham and Hitzler, 2014). In this context, a widely-adopted consensus-evaluation mechanism is based on majority voting, meaning that the most-selected option becomes the final task answer (Castano et al., 2015).

In CQ, we address crowdsourcing tasks by adopting the notion of **range task** (Bozzon et al., 2013). In a range task, the set of answers is not predefined and the worker executes the task by specifying a value belonging to a continuous interval of numbers. A well-known example of range task is described in Noronha et al. (2011) where crowdsourcing is proposed for estimating the amount of calories in a meal. In Noronha et al. (2011), a task is characterized by a picture of a dish and a worker receiving a task to execute is asked to insert a numeric value corresponding to her/his calorie estimation based on the given picture (min-max boundaries are provided to limit the range of possible values). On range tasks, majority voting mechanisms are ineffective since the worker answers can be distributed on a potentially-infinite range of values. An intuitive and popular solution for range task resolution is to employ a mean-based approach in which the arithmetic mean of the whole set of collected worker answers is provided as final result (Malone et al., 2010). However, in the literature, the use of a median-based approach to consensus evaluation in range tasks is considered as a preferable solution with respect to the mean-based ones (Galton, 1907). The adoption of a median-based approach is especially recommended when inaccurate or so-called "crank" workers (see Galton (1907)) can join the crowsourcing activities, which is a common situation in real crowdsourcing platforms.

A design choice of the proposed CQ approach is to rely on range tasks so that a crowdsourcing worker can express her/his mapping evaluations along the whole interval of values on which fairness is defined. The range tasks of CQ are characterized by i) the use of the interval $[0, 1]$ for enabling a worker to evaluate the fairness of a given mapping, and ii) the use of the *median-on-agreement* (ma) techniques for consensus evaluation on the collected task answers provided by crowdsourcing workers. The proposed ma mechanism aims to enforce a consensus mechanism where the answers of crank workers (i.e., workers with an outlier position) are not taken into account in determining the final task result.

Crowdsourcing techniques adopted in CQ are distinguished into *preparation* and *execution techniques* as described in the following.

4.1 Preparation techniques

Preparation techniques focus on the activity of a *requester*, namely the administrator of a crowdsourcing campaign, who has to configure the task setup before execution. Consider a set of tasks $\mathcal{T}$ to crowd-execute, where each task $T \in \mathcal{T}$ is a range task and it is associated with a corresponding mapping $m(i, j)$ to evaluate. A range task of CQ is defined as $T = \langle q, m, r, W, A, \bar{a} \rangle$ characterized by:

- a *task question* q providing a textual description submitted to workers for describing the activity to perform when executing the assigned task. For evaluation of mapping fairness, the adopted task question is "evaluate the similarity of the following entity descriptions".
- an *entity mapping* $m(i, j)$ providing a textual description of the entities i and j to consider linked through m;
- a *value range* $r = [min, max]$ denoting the range of $min - max$ numeric values that can be specified by crowd workers as task answer. For evaluation of mapping fairness, the adopted value range is $[0, 1]$;
- a *work force* $W = \{w_1, \ldots, w_k\}$ denoting the set of crowd workers involved in the execution of the task;
- an *answer set* $A = \{a_1, \ldots, a_k\}$ denoting the set of answers provided by the workers of W as result of task execution;
- a *final result* $\bar{a}$ denoting the final task result (i.e., the mapping fairness) determined according to the obtained answer set A.

The q, m, and r components are specified by the requester at design time, before crowdsourcing execution. In particular, each task $T \in \mathcal{T}$ is associated with a mapping and the $m(i, j)$ component is populated according to the features of the entities i and j. For the W component, the requester defines the size of the work force at design time (i.e., the parameter $k =\mid W \mid$). The size of the work force is fixed and stable for all the tasks $\mathcal{T}$ to execute. Then, the W component is progressively populated by the crowdsourcing platform at execution time. Each time a worker w asks for a task to execute, w is inserted into the work force of an available task (i.e., a task where the number of workers in W is less than the k parameter) and the task is assigned to w for execution[5]. When the worker $w_i \in W$ executes the task, the corresponding answer a_i is collected and it is inserted into the answer set A. When all the expected answers are inserted in A, meaning that the work force W is complete and all the workers in W provided their answers, a consensus evaluation mechanism is employed to determine the final task result $\bar{a}$ according to A (see Section 4.2).

Example. In Figure 3, we provide an example of range task T of CQ as it is shown to a worker w_i for execution. After the task question q, the entity mapping m is shown and it consists of a side-by-side presentation of the two instance descriptions linked by the mapping m. A slider is provided for enabling the worker to specify her/his own answer in the value range $r = [0, 1]$[6]. In the example, the worker w_i provides the answer $a_i = 7$. The button "Send the answer" is selected by the worker for insertion of a_i in the answer set A. The button "Reject the task" represents

[5]The criteria used for assigning tasks to workers are out of the scope of this work, and it depends on the specific task routing policies enforced by the crowdsourcing platform where the campaign is hosted.
[6]For the sake of clarity of crowd workers, the slider allows to specify an answer in the range [0, 10] which is eventually shifted to the range [0, 1] when the answer is inserted in A.

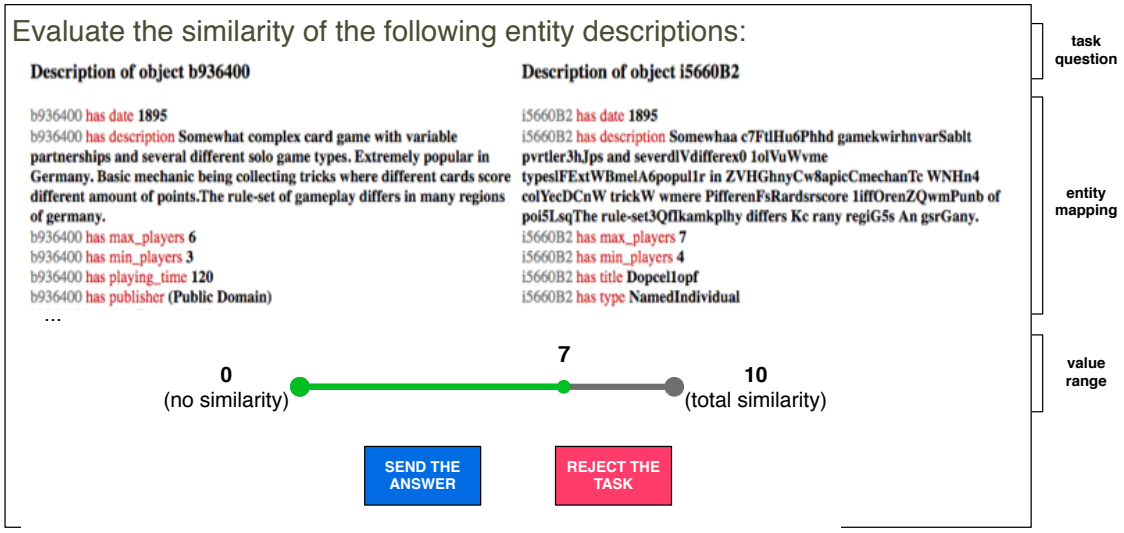

Figure 3 Example of range task of CQ shown to a worker w_i for execution

a possible worker option to skip the task execution when she/he is not confident to be able to provide a reliable answer. In this case, the task is assigned to another worker and the answer set A is not modified.

4.2 Execution techniques

Consider a range task T assigned to a group of workers $W = \{w_1, \ldots, w_k\}$ that provide a set of answers $A = \{a_1, \ldots, a_k\}$. In CQ, we rely on the median-on-agreement (ma) techniques to determine the final task result $\bar{a}$ (Genta et al., 2017). In a range task, each worker can provide a different answer value in the interval r, thus a fair solution to determine the task result is to calculate a "middlemost position" that represents a synthesis of the whole set of answers A. To this end, the ma techniques are characterized by i) the use of the median value to determine such a central position, and ii) the use of a consensus evaluation mechanism based on the *coefficient of variation* to distinguish the worker answers of A that express an agreement, from those that represent a discordant/outlier position. We call $W_C \subseteq W$ the *consensus group*, namely the subgroup of workers in W whose task answer in A has been recognized to express an agreement. We call $A_C \subseteq A$ the *consensus answers*, namely the set of task answers provided by the workers in W_C. Range task resolution according to the ma techniques is articulated in two main steps: *identification of the consensus group* and *definition of the task result* described as follows.

Identification of the consensus group. Consider the median value m_A calculated over the whole set of worker answers A. The consensus group W_C (and the specular set A_C of consensus answers) is built by iteratively considering the possible insertion of a worker w_i according to the distance from m_A of the related answer a_i, starting from the closest up to the most distant one. A worker w_i is inserted in W_C if the corresponding a_i value is "close enough" to the consensus answers already inserted in A_C. In other words, the consensus group is created by including workers that provided a similar answer, meaning that the answers of these workers denote a sort of agreement on the fairness to associate with the considered mapping. The *coefficient of variation cv* is exploited to decide whether a worker answer a_i is close enough, so that the worker w_i can be included in W_C. A threshold value th_{cv} is set by the requester at design time to decide whether the a_i value can be considered as similar to those previously inserted in A_C.

The identification of the consensus group is defined as follows (see Algorithm 1):

Data: Set of workers W and corresponding answers A, coefficient-of-variation threshold th_{cv}
Result: The task result $\bar{a}$
; // Compute the median m_A
$m_A \leftarrow median(A)$;
; // Identification of W_C
$W_C \leftarrow \emptyset$;
$A_C \leftarrow \emptyset$;
$cv \leftarrow 0$;
$a^* \leftarrow min_{a_j \in A}(\mid a_j - m_A \mid)$;
$w^* \leftarrow$ the worker that provided the answer a^*;
$A_C = A_C \cup a^*$;
$W_C = W_C \cup w^*$;
$A = A \setminus a^*$;
$W = W \setminus w^*$;
while $cv(A_C) \leq th_{cv}$ ***and*** $A \neq \emptyset$ **do**
  $a^* \leftarrow min_{a_j \in A}(\mid a_j - m_A \mid)$;
  $w^* \leftarrow$ the worker that provided the answer a^*;
  $A_C = A_C \cup a^*$;
  $W_C = W_C \cup w^*$;
  $A = A \setminus a^*$;
  $W = W \setminus w^*$;
end
$A_C = A_C \setminus a^*$;
$W_C = W_C \setminus w^*$;
$A = A \cup a^*$;
$W = W \cup w^*$;
; // Definition of $\bar{a}$
if $\mid W_C \mid > \mid W \mid$ **then**
  $\bar{a} \leftarrow m_{A_C}$;
end
else
  $\bar{a} \leftarrow null$;
end
return $\bar{a}$;

Algorithm 1: Algorithm for range task resolution according to ma

1. Compute the median m_A over the whole set of worker answers A and define $W_C = \emptyset$, $A_C = \emptyset$.
2. Select the worker answer $a^* \in A$ which is closest to m_A ($a^* = min_{a_j \in A}(\mid a_j - m_A \mid)$). Insert a^* in A_C and insert the worker w^* in W_C. Remove a^* from A.
3. Repeat the step 2 and evaluate if the last-inserted value a^* is similar to those previously inserted in A_C and it contributes to the formation of a consensus. To this end, calculate the coefficient of variation cv over the set of answers in A_C:

$$cv(A_C) = \frac{\sqrt{\frac{1}{|A_C|}\sum_{i=1}^{|A_C|}\left(a_i - \mu_{A_C}\right)^2}}{\mu_{A_C}}$$

where $|A_C|$ is the number of answers in A_C, a_i represents the i^{th} worker answer in A_C, and μ_{A_C} is the arithmetic mean of the answers in A_C.

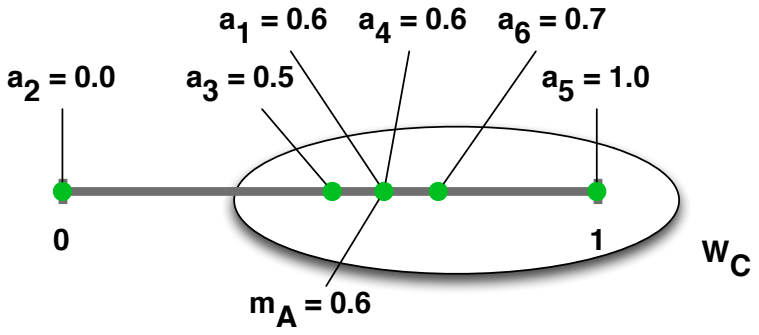

Figure 4 Example of task-result calculation according to the ma techniques

4. The threshold value th_{cv} is exploited to decide whether i) the last-inserted value a^* is confirmed in A_C ($cv(A_C) \leq th_{cv}$), or ii) it is removed from A_C ($cv(A_C) > th_{cv}$). In the latter case, the procedure for the construction of the consensus group is terminated.

Definition of the task result. A task is *committed* when the consensus group W_C contains the majority of workers of W, meaning that a valid consensus has been determined and the task evaluation is successfully completed. The final task result $\bar{a}$ is defined as the median value of the answers provided by the workers in the consensus group calculated over the set A_C, namely $\bar{a} = m_{A_C}$. $\bar{a}$ represents the fairness assessment of crowd workers for the considered task T and the associated mapping m. Otherwise, the task is *uncommitted*, meaning that the answers of the workers in W do not allow to recognize a consensus. In this case, the task result is unset, and the task can be scheduled for re-execution with a different work force. The maximum number of possible task re-executions due to uncommitted results is a parameter defined by the requester at design time of the crowdsourcing campaign. The higher is the number of possible task re-executions, the higher is the allowed crowdsourcing effort to reach a committed task result. When the maximum number of task re-executions has been reached and the result is still uncommitted, the task is *terminated*, meaning that the crowd cannot successfully determine the fairness of the mapping associated with the task.

Example. Consider the range task T of Figure 3. The task has been assigned to a work force composed of $k = 6$ crowd workers and the obtained answer set is $A = \{0.6, 0.0, 0.5, 0.6, 1.0, 0.7\}$. The median value over the whole set of worker answers is $m_A = 0.6$. The threshold for the coefficient of variation is set to $th_{cv} = 0.15$. By applying the Algorithm 1, we build the consensus group $W_C\{w_1, w_3, w_4, w_5, w_6\}$ shown in Figure 4. W_C represents a group of workers that agree on the fairness of the considered mapping, while the worker w_2 and the corresponding answer $a_2 = 0.0$ represents a disagreement with respect to the other workers of W and it is considered as an outlier position to discard. Since the majority of workers of W belong to W_C, the task is committed and the median value of the answers provided by workers in the consensus group is returned as final result of the task T: $\bar{a} = m_{A_C} = 0.6$.

4.3 Learning from the crowd

After the crowd execution, a set of mappings $\overline{\mathcal{E}} \subseteq \mathcal{E}$ is associated with a crowd-generated fairness degree. In particular, given $m_i \in \overline{\mathcal{E}}$, we call fairness degree $\rho(m_i)$ the final crowd result $\bar{a}$ for the task T_i associated with m_i. It is possible that the set $U = \mathcal{E} - \overline{\mathcal{E}}$ is not empty, meaning that mappings exist which are not associated with any fairness degree. In order to calculate the fairness degree of mappings in U, we exploit the annotated alignment $\overline{\mathcal{E}}^R$ in order to train a statistical model for predicting the crowd fairness degree for the mappings in U. In particular,

given the $t \times f$ matrix T^Q, where t is the size of the reference alignment $\mathcal{E}$ and f is the number of features used for assessing the quantity of transformation (see Section 3), we select the k rows corresponding to the mappings in $\overline{\mathcal{E}}$ as the training set for the learning model having the crowd fairness degrees $\overline{\mathcal{E}}^R$ as target. Learning is performed trough regression analysis to the goal of estimating the relationship between the quantity of transformation which generated the mappings and the crowd judgment. The model is then applied to U with the goal of automatically generating an estimated fairness degree for the mappings that have not been evaluated by the crowd.

5 Experimental results

For evaluation of the proposed CQ approach, we consider the mapgame case-study based on a dataset of entity mappings automatically generated through the SWING framework (Ferrara et al., 2011). A summary of the source ontology (Tbox and Abox) submitted to transformation is reported in Table 2, together with the summary of the target ontology resulting from the transformation.

Table 2 Summary of the source and the target ontologies used for producing the mapgame case-study

Summary of the source ontology

Classes	Object properties	Data properties	Individuals	DL expressivity
94	4	12	3,731	$\mathcal{AL}(D)$
Class assertions	**Object property assertions**	**Data property assertions**	**Boardgames**	**Videogames**
9,749	2,937	18,510	2,031	1,037

Summary of the target ontology

Classes	Object properties	Data properties	Individuals	DL expressivity
94	4	12	7,008	$\mathcal{AL}(D)$
Class assertions	**Object property assertions**	**Data property assertions**	**Boardgames**	**Videogames**
4,269	3,875	15,754	1,342	545

The case-study consists of a set of 3 731 mappings involving different levels and types of transformations on 16 properties used for describing individuals representing board- and video-games. The quantity of transformation T_i^Q for each mapping m_i has been assessed according to the following scheme. Since there are multiple transformations per property, and several transformations can be applied simultaneously per instance, analyzing the transformations independently would yield misleading results. To account for this, we grouped transformations by property, and used a simple system to score the quantity of transformation T^Q:

- 1 - if no transformation was applied to the property for that instance
- 0.3 - if the value of the property was split or the property was reified and there was no value change, under the rationale that the value of the property is still present, but much harder to find
- 0 - if the property was deleted or the value for the property was edited

While it may seem unintuitive that we did not distinguish between cases of deletion and edition, we observed that changing even 5% of a string randomly is sufficient to make it unrecognizable, and any change to a numeric value has the same effect. Thus, edition is arguably even worse than deletion for a matching system, as it not only impedes correct matching, but can also lead to false positives.

Table 3 Evaluation of the results of AML and LogMap based on the reference alignment derived from the transformation process.

System	TP	FP	FN	Precision	Recall	F-measure
AML	1,407	408	2,324	77.5%	37.7%	50.7%
LogMap	1,438	694	2,293	67.4%	38.5%	49.0%
intersection	1,300	99	2,431	92.9%	34.8%	50.7%
union	1,545	1,003	2,186	60.6%	41.4%	49.2%

Given the mappings for mapgame and its associated quantity of transformation T^Q, the evaluation has been performed in five steps: i) we initially executed two state of the art matching tools, namely LogMap (Jiménez-Ruiz and Cuenca Grau, 2011; Jiménez-Ruiz et al., 2012a) and AgreementMakerLight (AML) (Faria et al., 2013), in order to evaluate their results before the mapping fairness evaluation; ii) then, we enforced a crowdsourcing campaign to measure the fairness of each mapping; iii) we assessed the quality of the mappings in terms of fairness by also setting up a threshold for mapping refinement; iv) we applied linear regression to extend the crowd evaluation to all the mappings in the reference alignment of mapgame; v) finally, we studied how the refinement of the reference mappings will change the performances of the tools in terms of Precision and Recall.

5.1 Matching Systems

LogMap (Jiménez-Ruiz and Cuenca Grau, 2011; Jiménez-Ruiz et al., 2012a) and AgreementMakerLight (AML) (Faria et al., 2013) are state-of-the-art ontology matching systems with a long track record of performance in the Ontology Alignment Evaluation Initiative. Both systems are highly versatile and scalable, handling a wide range of ontology and instance-matching problems, with tens (or even hundreds) of thousands of entities to match. Both also feature logical repair algorithms, which differ in that LogMap attempts to minimize violations of the consistency and locality principles (Jiménez-Ruiz et al., 2011) whereas AML is only concerned with the former.

Both LogMap and AML are primarily lexical-based systems, relying mainly on the similarity between the vocabularies of the input ontologies (i.e., the annotations and/or data property values) to match entities. They differ slightly in that LogMap uses all lexical information together, whereas AML separates annotations and given names (i.e., values of data properties such as "title" or "name") from other data property values, and does not use the former in its default instance matching pipeline.

There are several other instance matching/data linking systems (e.g. Legato (Achichi et al., 2016), SILK (Volz et al., 2009), and LIMES (Ngomo and Auer, 2011)). However, AML and LogMap were chosen for their good performance in instance matching tracks at the OAEI, code availability and also familiarity to the authors.

5.2 Results

Table 3 shows the results for LogMap and AML, as well as the intersection and union of their alignments. As expected given the nature of the transformations, the two systems have a relatively poor performance, with an F-measure of only around 50%. AML has a higher precision but lower recall than LogMap, which is at least partially due to the fact that it does not use the titles to perform matching. The intersection and union results show that the two systems agree on most of the mappings they predict right and disagree on most of those they predict wrong, having a Jaccard index of 84% for the true positives but only 10% for the false positives (see also Figure 5).

In an effort to understand the effect of the different types of transformations on the matching ability of the two systems, we wanted to correlate transformations with mapping status (i.e.,

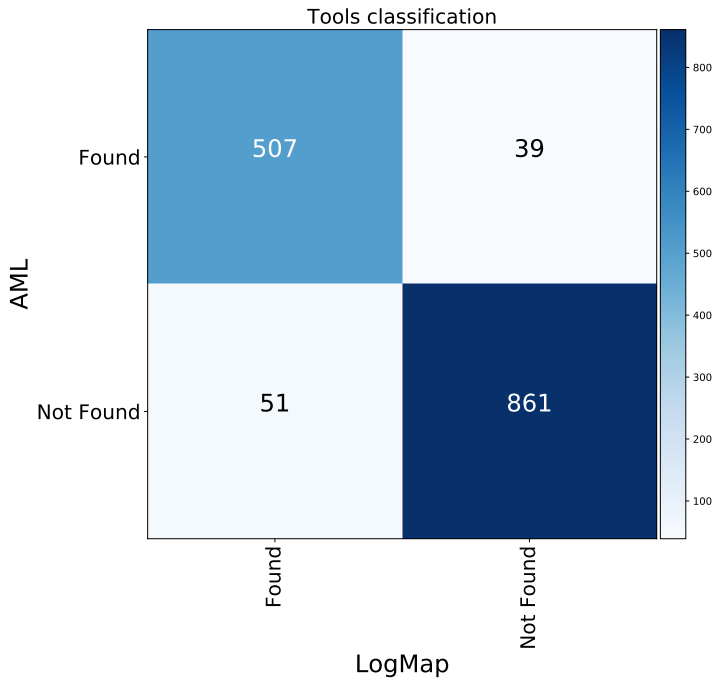

Figure 5 Confusion matrix of the AML and LogMap results

whether the system was able to find the mapping). Following this scoring T^Q presented above, we computed Pearson's correlation coefficient between mapping status and property integrity for LogMap, AML, and their union and intersection. The results, shown in Table 4 reveal that there is not a single property highly correlated with the mapping status, but rather a number of weak correlation relations between properties and mappings. On one side, this suggests that matching tools produce the decision on a mapping by combining information coming from different property values in combination. On the other side, however, we note some differences among properties and their correlation with mappings. In particular, "description" is the property more correlated with the ability of the systems to find the transformation. "Title" also has a somewhat meaningful correlation for LogMap, but not AML, as the latter is not using this property to perform matching. The remaining properties have very low correlation coefficients.

We complemented this analysis by doing a linear regression between mapping status and property integrity for the union of LogMap and AML. These results, shown in Table 5 confirm that "description" is the property that most affects matchability, followed by "title" and "publisher". No other property has a statistically significant linear coefficient. The pattern behind these results is quite evident: "description", "title" and "publisher" are all string-valued properties, with extensive coverage (they are the three textual properties with the greatest coverage, at over 60% of the individuals each) and great variety in values (title is unique per game individual, description is unique at least per family of games, and publisher is quite diverse including some unique values). Thus, these three properties are the best sources of information to use in matching individuals. The reason why "description" has a higher impact on matchability than "title" even for LogMap is likely due to the fact that there are very few transformations with preserved "title" (84) while a fair number have preserved "description" (557).

A manual analysis of the matching results of the two systems showed that both systems failed to find the correct mapping in presence of the "description" only when the description was not unique, but shared among several games in a family (e.g., the several "Monopoly" games).

Table 4 Correlation between property integrity and ability of the matching systems to find match the transformation for AML, LogMap, and their intersection and union.

Property	AML	LogMap	Intersection	Union
Description	0.34	0.27	0.29	0.32
Title	-0.02	0.11	-0.01	0.10
Playing Time	0.03	0.05	0.03	0.05
Publisher	0.03	0.01	0.01	0.02
Market	0.01	0.02	0.01	0.02
Min Players	0.01	0.03	0.02	0.02
Made For	0.01	0.03	0.03	0.01
Device	0.01	0.01	0.01	0.01
Date	0.00	0.00	0.00	0.01
Type	0.02	0.00	0.01	0.01
Rated	0.00	-0.01	-0.01	0.00
Source	-0.01	0.00	-0.01	0.00
Price	-0.02	-0.01	-0.02	0.00
Vote	0.00	0.00	0.00	-0.01
Vendor	-0.01	-0.01	-0.01	-0.01
Max Players	-0.02	-0.02	-0.02	-0.02

Table 5 Coefficients and respective p-values for the linear regression between mapping status in the union of LogMap and AML and property integrity for all properties.

Property	Coefficients	p-value
Description	0.44	0.00
Title	0.32	0.00
Publisher	0.11	0.00
Price	-0.03	0.16
Type	0.03	0.17
Rated	0.02	0.22
Playing Time	0.02	0.35
Date	0.01	0.40
Source	-0.01	0.65
Device	0.01	0.73
Max Players	-0.01	0.73
Min Players	0.01	0.83
Vote	0.00	0.88
Market	0.00	0.90
Vendor	0.00	0.91
Made For	0.00	0.97

5.3 Crowd-evaluation of mapping fairness

A crowdsourcing campaign has been enforced to crowd-evaluate the fairness of mapgame mappings according to the ma techniques presented in Section 4. For task execution in the campaign, we employed the **Argo crowdsourcing system** where the ma techniques have been implemented.[7] Each mapping in mapgame corresponds to a task in the crowdsourcing campaign. Moreover,

[7]Further details about Argo and related crowdsourcing techniques for consensus evaluation are provided in Castano et al. (2016).

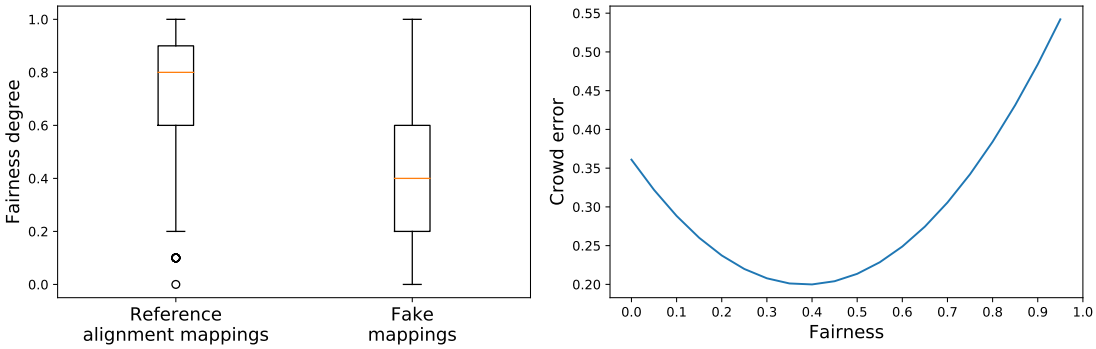

Figure 6 Distribution of crowd judgments and crowd-error function

according to the procedure described in Section 3, a set of 697 "fake mappings" tasks has been added to the tasks of the crowdsourcing campaign. The crowdsourcing activities of the mapgame case-study were performed from November 10th, 2018 to December 10th, 2018 by relying on a crowd of 163 workers selected from a class of master-degree students (average worker age is 21 years old). For task resolution, we asked the workers to rely on their personal knowledge and we set the available time to perform a task to a maximum of 15 minutes. Crowdsourcing tasks have been configured with a work force of 6 different crowd workers randomly selected from the available pool of workers. A threshold $th_{cv} = 0.15$ was specified in Argo for identification of the consensus group and consensus evaluation of range-task results according to the ma techniques.[8] The number of committed tasks was 2,155 of which 1,458 mappings are taken from the reference alignment. The crowd judgment on reference and fake mappings is shown in Figure 6, as well as the crowd-error function derived from the mapgame experiment.

By analyzing the distribution of the fairness degrees assigned to the mappings, we observe that the crowd workers are generally capable of recognizing high fairness degrees to the mappings from the reference alignment, and low fairness degrees to the fake mappings. However, by focusing on the mappings from the reference alignment, we note that the fairness degree spans from 0.6 to 0.9, meaning that some transformations produced mappings that are less fair than others according to the crowd judgement. To determine the most appropriate threshold value for categorically deciding which mappings are actually fair, we analyze the crowd-error function that measures the error of the crowd in relation with different fairness degrees. As a result, a threshold $th = 0.4$ is chosen which corresponds to the minimum value of the error function (error value around 0.2).

As a final observation on the crowd results, we compared the quantity of transformation applied to the mappings in the reference alignment against the fairness results provided by the crowd and we trained a linear regression model to predict the fairness judgments for the mappings that the workers did not evaluate during the crowdsourcing activities (see Figure 7).

In order to analytically compare the quantity of transformation with the fairness provided by the crowd, we exploited the T^Q matrix where rows represent mappings that have been evaluated by the crowd and columns represent the quantity of transformation grouped by the 16 properties associated with the source ontology individuals. We recall that the values of T^Q are inversely proportional to the quantity of transformation, with the value 1 representing the fact that the assertions for a specific property are identical in the source and the target ontology. Thus, we synthetically represent the quantity of transformation by a measure of the inverse cumulative quantity of transformation iqt_i for each mapping vector $\vec{q_i}$ of length N_q in T^Q. The measure iqt_i

[8]The value of the threshold th_{cv} has been determined on the basis of experimental observations to maximize the trade-off between the number of committed tasks (i.e., tasks with successful consensus evaluation) and the number of worker answers to consider in the consensus group (see the discussion on the ma techniques provided in Section 4.2).

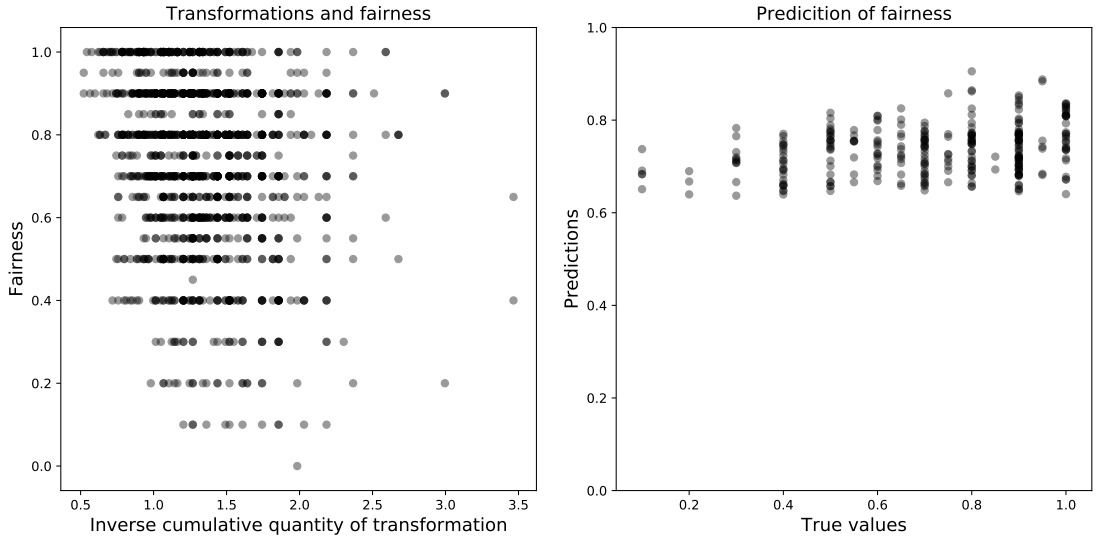

Figure 7 Regression analysis on crowd fairness judgments

is defined as:

$$iqt_i = \log\left(\frac{N_q}{\sum_{j=1}^{N_q} q_i[j]}\right)$$

Figure 7 shows how the crowd provided high values of fairness for mappings affected by low levels of *iqt* (*iqt* lower than 1.0), as expected. However, there is also a number of mappings with high values of *iqt* that have been judged as fair (fairness higher than the threshold 0.4). This confirms the idea that when transformations are not applied to specific properties that are crucial for identifying entities (i.e., Description and Title in our experiment) the crowd is still capable of recognizing the entity correspondence. In order to learn a predictive model for the crowd judgment, we enforce linear regression using the 1,458 mappings evaluated by the crowd as the training set, where their quality of information vectors represent the features and the crowd judgment the target variable. We run also tests using other regression models, including polynomial models, but we obtained the best results with linear regression, which has a mean squared error equal to 0.03.

In Figure 7, we compare the fairness values predicted by the model with the true fairness values provided by the crowd. In general, the model overestimates fairness for mappings with a low crowd fairness, due to the peculiar distribution of crowd fairness judgments where the mappings with fairness lower than 0.4 are few. For fair mappings instead, the model performs well in predicting the crowd judgment.

Evaluation of the ma techniques. As a further experiment on the crowdsourcing results, we analyzed the reliability of the ma techniques as a mechanism for consensus evaluation of range tasks. We compared the fairness values obtained through the median-on-agreement techniques against a conventional *majority-based mechanism* applied to the task answers provided by crowd workers. The majority-based mechanism is based on the use of boolean tasks where crowd workers can provide only a *true/false* answer. In particular, for a task T and corresponding mapping $m(i, j)$ to evaluate, the possible answers of a worker w_k can be i) $a_k = 1$, meaning that w_k considers $m(i, j)$ as a fair mapping (according to the worker w_k, it is *true* that the mapping is fair), or ii) $a_k = 0$, meaning that w_k considers $m(i, j)$ as an unfair mapping (according to the worker w_k, it is *false* that the mapping is fair). According to the majority-based mechanism, the

mapping $m(i, j)$ of a task T is fair when the number of collected *true answers* is higher than the number of collected *false answers* within the work force W involved in the task execution. Otherwise, $m(i, j)$ is evaluated as an unfair mapping. To enforce our experiment, we converted the answers provided by crowd workers to range tasks into boolean answers. We call $a_k^m \in [0, 1]$ the answer provided by a worker w_k to a range task in the crowdsourcing campaign. We call $a_k^b \in \{0, 1\}$ the corresponding boolean answer obtained from a_k^m according to the following function:

$$a_k^b = \begin{cases} 1, & \text{if } a_k^m \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

Given a task T, the *majority-boolean task result* $\bar{a}^{mb}$ is determined by calculating the majority of true/false answers a^b within the set of worker answers to the task T. According to the boolean model, the fairness of a mapping $m(i, j)$ associated with a task T can be $\bar{a}^{mb} = 1$ ($m(i, j)$ is evaluated as a fair mapping) or $\bar{a}^{mb} = 0$ ($m(i, j)$ is evaluated as an unfair mapping). Consider the range task result $\bar{a}$ determined though the ma techniques presented in Section 4. For the sake of our experiment, we convert the range task result $\bar{a}$ into a *converted-boolean task result* $\bar{a}^{cb}$. Then, for each task T, we compare the converted-boolean task result $\bar{a}^{cb}$ against the majority-boolean task result $\bar{a}^{mb}$. As a result, we observe that the two approaches to consensus evaluation provide equivalent results and they differ in just 0.3% of task evaluations. Such a result is not surprising since informed and motivated workers have been selected for participation to the crowdsourcing activities. The benefit of adopting ma with respect to conventional consensus evaluation mechanisms becomes evident when outlier positions due to inaccurate workers need to be managed. To this end, we enriched the "pure" crowdsourcing answers with additional "crank" answers with the aim to simulate the presence of inaccurate workers in crowdsourcing participants. These additional answers are around 30% of crowdsourcing answers and they are uniformly distributed over the tasks. The answer value is set to represent an outlier worker position (i.e., distant from the committed task answer $\bar{a}$). In presence of crank workers with corresponding outlier positions, for each task T, the comparison of converted-boolean task result $\bar{a}^{cb}$ against the majority-boolean task result $\bar{a}^{mb}$ shows that the two approaches differ in around 3.5% of task evaluations. This is a confirmation that differences in the two consensus evaluation approaches increase as long as crank answers appear. We also analyzed the task results of each consensus evaluation approach when "pure" answer and "pure + crank" answers are considered. In converted-boolean task results, which are based on ma, the introduction of crank answers causes the change of $\bar{a}^{cb}$ result in around 1.4% of tasks. This means that the final task result is actually affected by the crank answers in around 1.4% of tasks. In majority-boolean task results, the introduction of crank answers causes the change of $\bar{a}^{mb}$ result in around 4.1% of tasks. As a result, we confirm that the ma techniques provide more reliable results than majority-based approach in consensus evaluation, and this is especially true when a certain number of crank workers participate in the crowdsourcing activities.

5.4 Mapping and evaluation refinement

Given the crowd results, the final step of our experiment was devoted to refine the initial reference alignment $\mathcal{E}$ in order to evaluate the matching tools by taking into account the mapping fairness. As a first step, we exploit the predictive model to extend the crowd judgment to all the mappings in $\mathcal{E}$, including those that have not been evaluated by the crowd. All the 3,731 mappings in $\mathcal{E}$ are now associated with a fairness degree, but the mappings retrieved by the tools that are not part of $\mathcal{E}$ (i.e., false positives) are not associated with a measure of fairness. Thus, instead of performing a new evaluation of the tools based on Precision and Recall, we take into account only $\mathcal{E}$ and we perform and evaluation of sensitivity by comparing the set $TP^t \subseteq \mathcal{E}$ of reference mappings retrieved by a tool t (i.e., true positives) with the set $\mathcal{E} = \mathcal{E} - TP^t$ of reference mappings that have not been retrieved by t (i.e., false negatives). In particular, we define three measures of

Table 6 Sensitivity after the refinement of reference mappings

Tool	σ	σ_w	σ_r
AML	0.38	0.41	0.39
LogMap	0.39	0.42	0.40
Intersection	0.35	0.38	0.36
Union	0.41	0.45	0.43

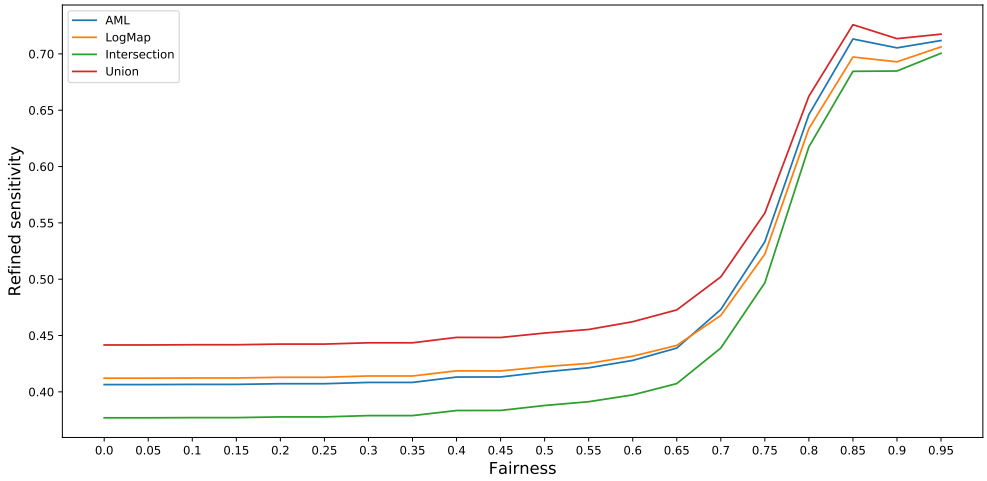

Figure 8 Refined sensitivity at different levels of fairness

sensitivity, namely baseline sensitivity σ, weighted sensitivity σ_w, and refined sensitivity σ_r as follows:

$$\sigma = \frac{|\,TP^t\,|}{|\,\mathcal{E}\,|}, \quad \sigma_w = \frac{\sum_{i=1}^{|TP^t|} r_i}{\sum_{j=1}^{|\mathcal{E}|} r_j}, \quad \sigma_r = \frac{|\,\{m_i \in TP^t : r_i > th\}\,|}{|\,\{m_j \in \mathcal{E} : r_j > th\}\,|}$$

The baseline sensitivity σ is a measure of the instance matching sensitivity before the mapping refinement. The weighted sensitivity σ_w measures sensitivity by taking into account the fairness degrees provided by the crowd. Finally, the refined sensitivity σ_r exploits a categorical classification of mappings by taking into account only the mappings that are considered highly relevant according to the crowd, that are those having a fairness higher then the threshold of 0.4. The results of evaluation refinement according to sensitivity is reported in Table 6.

The refinement of reference alignment mappings improves the sensitivity performances of the tools, including the intersection of union of their results, in all the cases. This is relevant in the evaluation process in that we aim at achieving a "fair evaluation". The notion of evaluation fairness is based on the idea that a matching tool should be able to reproduce the human judgment about the similarity between different ontological descriptions of the same objects. The activity of evaluating this capability is fair when the mappings to be retrieved are actually detectable for a human expert. On the contrary, when we ask a tool to retrieve a mapping that is not detectable neither for a human expert, we are actually asking the tool to perform a task that is not achievable, which is unfair. With `CQ`, the mapping refinement takes into account only the tools results that are considered fair from the crowd. Thus, the interesting point in the new values of sensitivity is not that they are higher, but rather that they are more fair because they are calculated on mappings for which the transformations do not impede manual detection.

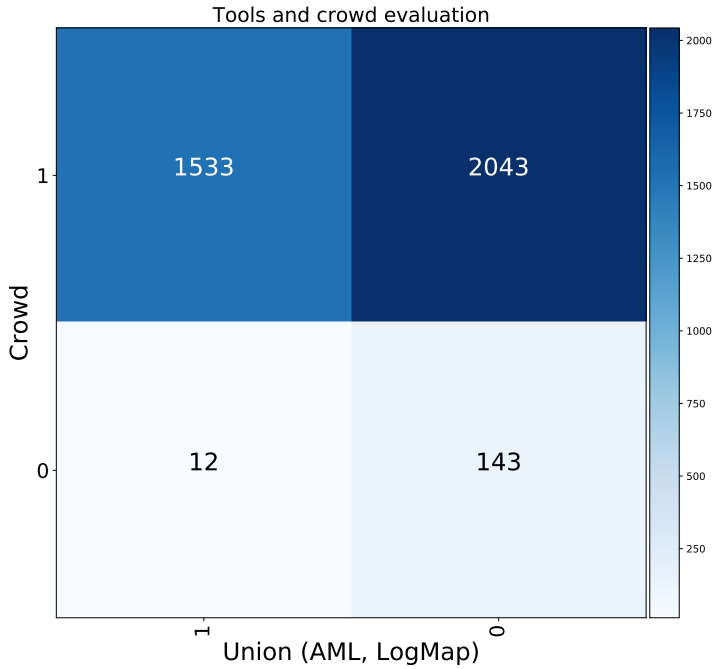

Figure 9 Tools and crowd evaluation

Figure 8 describes the behavior of refined sensitivity at different levels of fairness, which is similar for AML and LogMap. In particular, we note that when we choose a value greater than 0.7 of fairness, the sensitivity of the tools increases remarkably. This suggest that 0.7 is a critical value for detecting those mappings that are very simple to detect. We have already seen that mappings under 0.4 of fairness should be pruned from the reference alignment. Thus, the critical range for evaluation in our experiment was between 0.4 and 0.7, where we find mappings quite fair according to the crowd but not retrieved as matching by the tools, while the number of mappings classified as correct by the tools but not by the crowd is very limited (see Figure 9).

6 Related work

Ontology alignment systems are typically evaluated using a dataset composed by two ontologies, source and target, and a reference alignment between them. The alignment produced by the system is compared to the reference, and Precision and Recall are computed. Ideally, reference alignments would be produced by a group of domain experts that would reach a consensus on how to align the two ontologies. However, given the cost of producing reference alignments manually, there have been a number of automated approaches proposed in the last years.

In 2009, the Ontology Alignment Evaluation Initiative (OAEI) introduced the Instance Matching track which used an automatically generated benchmark dataset, IIMB.[9] The benchmark applies transformations to a source ontology to produce a target ontology. This benchmark was also used in the subsequent years. In 2013, the OAEI IM track employed the RDTF tool to generate a benchmark that includes controlled transformations over the source data, namely value, structural and translations (Grau et al., 2013). The following year, the sub-task of identity recognition also employed datasets based on automated modifications over the source. In 2015, the OAEI introduced SPIMBENCH (Saveta et al., 2015), which is applicable to RDF data with an associated schema. SPIMBENCH also employs transformations, but in addition it provides

[9]http://islab.di.unimi.it/iimb/

a scalable data generator, a reference alignment, and evaluation metrics. In 2016, in addition to SPIMBENCH, the OAEI also introduced a new dataset generated using UOBM and transformed using LANCE (Achichi et al., 2016). In 2018, IIMB was once again part of the OAEI. For IIMB, each IM task has been created by systematically applying a set of transformations to the source ontology. The TBox is unchanged, while the ABox is altered in several ways by transformation operations, namely data value transformation, data structure transformation, and data semantics transformation.

One of the challenges automated benchmark generation encounters is scalability. The HOBBIT platform (Röder et al., 2017) aims at benchmarking Big Linked Data systems. It can handle benchmarks that use single consecutive requests, but also benchmarks that have a high workload through parallel requests. Automated generation of benchmarks through transformations over a source ontology is the most common technique showcased by successive IM tracks at OAEI. There are also other techniques that can be employed, for instance adapting external resources that have mappings to the source and target ontologies (Jiménez-Ruiz et al., 2012b).

In recent years, crowdsourcing has become popular within the Semantic Web community for a variety of tasks. For instance, the use of crowdsourcing has been proposed for supporting the ontology engineering process where the crowd can be effectively employed to enforce ontology verification. In Mortensen (2013), crowdsourcing is exploited to detect *extralogical errors* (i.e., non-logical errors than can only be detected through human interpretation) as part of a comprehensive framework for Quality Assurance of ontologies at scale. As a further example, the specification of both system and methods for integrating the use of crowdsourcing in a platform for fuzzy concept mapping is described in Van Dusen et al. (2016). About ontology matching, a natural context where crowdsourcing can be employed is in interactive ontology matching tools. In Paulheim et al. (2013), the proposed framework is based on the idea to involve users in the evaluation of the ontology mappings generated by matching tools. The idea is to rely on user expertise to address the resolution of complex mappings. In this respect, the user collaboration contributes to provide feedback about the tool performances and it is useful to detect conflicts and to enforce decision support in controversial situations. However, this example of interactive matching process is more similar to a collaboration-oriented framework rather than to a crowdsourcing-oriented platform since the basic crowdsourcing features, such as for example the worker independence, the massive involvement of a crowd without any qualified expertise, and the presence of a rewarding mechanism, are not considered/supported.

Mostly, the role of crowdsourcing in ontology matching is related to mapping validation. In this direction, a discussion regarding user alignment validation is provided in Dragisic et al. (2016) and Li et al. (2019). The authors stress the importance of involving a group of *informed users* to enforce mapping validation, so that they are familiar with ontologies and related formal representations, which is in contrast with the basic crowdsourcing principles. The comparison of an expert-based mapping validation against a crowd-based one is provided in Acosta et al. (2013) with a case-study in context of the linked data cloud. The goal of Acosta et al. (2013) is to understand the possible contributions of experts and crowdsourcing in the resolution of ontology matching issues, respectively. A similar experiment is discussed in Noy et al. (2013). A case-study in the context of ontology engineering is presented to compare the quality of results provided by crowdsourcing workers in the Amazon Mechanical Turk platform (AMT) and undergraduate students recruited according to expertise-based criteria. The authors claim that comparable results are provided by the crowd and the students, thus confirming the idea that a positive contribution can be provided by crowdsourcing, especially on the large scale.

In Sarasua et al. (2012), the *CrowdMap* model is presented where real crowdsourcing models and techniques are employed in validation of mappings discovered by automatic tools. A similar solution is proposed in Cruz et al. (2014) where mapping validation is enforced through a *pay-as-you-go* approach characterized by the use of crowd feedback. A propagation mechanism is also presented in Cruz et al. (2014) to reduce the number of mappings to validate by applying the

feedback-based result obtained on a specific mapping to other similar mappings. In both Sarasua et al. (2012) and Cruz et al. (2014), multiple user feedbacks are collected for a task and a consensus needs to be reached to validate or to reject a given mapping. On this point, it is important to note that a boolean task model is employed in both the proposed solutions. This means that, receiving a mapping to validate, the user/worker can only answer with a boolean reply where she/he evaluates if the mapping is "correct" or "incorrect". The opportunity to rate the degree of correctness or incorrectness is not provided.

The possible role of crowdsourcing in the creation of ontology matching benchmarks is introduced in Cheatham and Hitzler (2014). In particular, the authors stress the impossibility to rely on experts for fully validating the contents of a (large) automatically-generated benchmark. At the same time, the authors present an alternative approach based on AMT and crowdsourcing for successfully scaling on large mapping benchmark validations. Again, a boolean task model is adopted and workers are asked to evaluate mappings by distinguishing correct and incorrect ones. In Thaler et al. (2011), a game for ontology alignment is presented called *SpotTheLink*. The authors argue that ontology matching requires a human input for improving the quality of results provided by automated tools. Such an input can be harvested as a side-product of an entertaining collaborative online game. The game is articulated into a series of challenges, where a challenge (i.e., a task) is a mapping to validate/confirm. A player/worker is asked to choose the most appropriate mapping for a given entity by selecting her/his answer among a set of possible alternatives. The challenge is successfully completed only if a majority of the involved workers provide the same answer. The outcomes of a challenge execution are twice. On one side, a successfully completed challenge represents a confirmed/validated mapping on which a consensus has been reached. On the other side, the players are rewarded with an increasing score when they participate in the consensus. As a side-effect, the top-score players are also the top-rated ontology validators. In Thaler et al. (2011), a game challenge represents a crowdsourcing task based on a *choice answer*. The use of a choice-task model can be considered as an improvement of the boolean-task model adopted in Sarasua et al. (2012) and Cruz et al. (2014). However, the opportunity to rate the quality of a mapping is still not supported.

Original contribution. With respect to the literature solutions discussed above, a first contribution of the proposed CQ approach is the capability to trace the relation between the effect of a transformation on a mapping and its impact on the possibility of a human to provide a correct interpretation of that mapping. Mappings are boolean relationships, so transformation processes need to define a boundary between transformation effects that still lead to a valid mapping and those that do not. In CQ, we propose to rely on human judgements to bridge this gap and we present crowdsourcing techniques based on range tasks and consensus mechanisms to extend the conventional boolean/choice model adopted in the literature. Related work approaches (e.g., Sarasua et al. (2012); Cruz et al. (2014); Cheatham and Hitzler (2014)) involve crowd workers in tasks where only "valid" or "non-valid" options are available as possible answers. The crowdsourcing techniques of CQ are characterized by the use of range tasks to enable a crowd worker to really rate/evaluate the quality of an automatically-generated mapping by associating a measure of fairness in a continuous range of possible values, thus allowing a more fine-grained evaluation than the boolean one. For enforcing a robust crowdsourcing evaluation that effectively expresses the human judgement of the mapping fairness, consensus-based techniques are employed to measure the agreement of workers that execute the same task. This is particularly challenging in range tasks, since different workers can provide different ratings in a continuous range of values for the same mapping. On this point, the contribution of CQ is the use of the ma (median-on-agreement) techniques for extending conventional crowdsourcing solutions suited for boolean/choice tasks. In particular, we propose the adoption of a majority-based mechanism specifically conceived to deal with range tasks that is capable of distinguishing the worker answers that express an agreement, from those that represent a discordant/outlier position.

Finally, we note that scalability issues are usually mentioned as a limitation in manual, expert-based approaches to mapping evaluation of automatically-generated benchmarks. As described in Cheatham and Hitzler (2014), crowdsourcing can help to mitigate the impact of such kind of issues. However, budget constraints are not considered and it is possible that the size of the generated benchmark is greater than the available budget to use for rewarding the crowd of workers to involve in mapping evaluation. A further contribution of CQ is the use of a supervised learning model so that the results of crowd-assessments on a budget-compatible number of mappings are propagated to the whole dataset of mappings to evaluate by exploiting the kind and the degree of transformations applied during the benchmark generation.

7 Concluding remarks

Reference alignments are crucial to support the evaluation of data linking methods, but their manual creation is very time-consuming and typically relies on domain expert knowledge. Synthetic approaches to produce reference alignments based on transformations of a source dataset have been applied to overcome this issue, however, the automated application of transformations can produce unfair mappings, i.e., mappings for which the transformation to produce the target entity was disruptive enough to render a mapping too hard to find, and thus unfair to be used in evaluation. We have proposed a crowd-based approach to assess mappings fairness, which is applied to the refinement of synthetic reference alignments to make them more useful and fair. Our approach is based on the agreement of crowd workers over the fairness of a mapping and can be applied to larger datasets through supervised learning. By being able to separate mappings with lower fairness from those with a higher fairness, the evaluation of systems becomes more fair, since the reference alignment is only composed of detectable mappings, i.e., mappings for which the transformations do not impede their manual detection.

We have run experiments on an instance matching task, and assessed the impact of using the refined reference alignments on the evaluation of two state of the art instance matching systems. Our experiments show that there is a range of fairness for which crowd workers are able to detect correct mappings, but where automated systems struggle. These more challenging mappings are crucial for system evaluation, and represent the current blind spots of systems, and can thus be helpful in driving systems innovation and development.

Possible extensions of the CQ approach are about the adoption of customization techniques for dynamically selecting the most appropriate work force of a given crowdsourcing task according to the complexity of the mapping to evaluate based on the number of rejected assignments and on the answer quality provided in previously-executed tasks. Moreover, we aim at running further experiments with different datasets. In particular, we plan to enforce machine learning techniques to generalize the notion of fairness from the crowd and check how much mapping fairness depends on the specific dataset used for evaluation.

Acknowledgement

Daniel Faria was funded by the EC H2020 grant 676559 ELIXIR-EXCELERATE and the Portuguese FCT Grants 22231 BioData.pt (co-financed by FEDER) and UIDB/50021/2020 to INESC-ID). Catia Pesquita was supported by FCT through the LaSIGE research unit (ref. UIDB/00408/2020 and ref. UIDP/00408/2020) and project SMILAX ref. PTDC/EEI-ESS/4633/2014. Ernesto Jiménez-Ruiz was supported by the AIDA project, The Alan Turing Institute under the EPSRC grant EP/N510129/1 and the SIRIUS Centre for Scalable Data Access (Research Council of Norway, project 237889).

References

Achichi, M., Cheatham, M., Dragisic, Z., Euzenat, J., Faria, D., Ferrara, A., Flouris, G., Fundulaki, I., Harrow, I., Ivanova, V., et al. (2016). Results of the ontology alignment evaluation initiative 2016. In *OM: Ontology matching*, pages 73–129. No commercial editor.

Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., and Lehmann, J. (2013). Crowdsourcing Linked Data Quality Assessment. In Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J. X., Aroyo, L., Noy, N., Welty, C., and Janowicz, K., editors, *Proc. of the 12th Int. Semantic Web Conference*, pages 260–276, Sydney, Australia.

Algergawy, A. et al. (2018). Results of the ontology alignment evaluation initiative 2018. In *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference*, pages 76–116.

Arolas, E. E. and de Guevara, F. G.-L. (2012). Towards an Integrated Crowdsourcing Definition. *Journal of Information Science*, 38(2):189–200.

Bozzon, A., Brambilla, M., Ceri, S., and Mauri, A. (2013). Reactive Crowdsourcing. In *Proc. of the 22nd Int. World Wide Web Conference (WWW 2013)*, pages 153–164, Rio de Janeiro, Brazil.

Carmines, E. G. and Zeller, R. A. (1979). *Reliability and validity assessment*, volume 17. Sage publications.

Castano, S., Ferrara, A., Genta, L., and Montanelli, S. (2016). Combining Crowd Consensus and User Trustworthiness for Managing Collective Tasks. *Future Generation Computer Systems*, 54.

Castano, S., Ferrara, A., and Montanelli, S. (2015). A Multi-Dimensional Approach to Crowd-Consensus Modeling and Evaluation. In *Proc. of the 34th Int. Conference on Conceptual Modeling (ER 2015)*, Stockholm, Sweden.

Cheatham, M. and Hitzler, P. (2014). Conference v2.0: An Uncertain Version of the OAEI Conference Benchmark. In Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., and Goble, C., editors, *Proc. of the 13th Int. Semantic Web Conference*, pages 33–48, Riva del Garda, Italy.

Cruz, I. F., Loprete, F., Palmonari, M., Stroe, C., and Taheri, A. (2014). Pay-As-You-Go Multi-user Feedback Model for Ontology Matching. In *Proc. of the 19th Int. Conference on Knowledge Engineering and Knowledge Management*, pages 80–96, Link'oping, Sweden.

Dragisic, Z., Ivanova, V., Lambrix, P., Faria, D., Jiménez-Ruiz, E., and Pesquita, C. (2016). User Validation in Ontology Alignment. In *Proc. of the 15th Int. Semantic Web Conference*, Kobe, Japan.

Euzenat, J., Rosoiu, M., and dos Santos, C. T. (2013). Ontology matching benchmarks: Generation, stability, and discriminability. *J. Web Semant.*, 21:30–48.

Euzenat, J. and Shvaiko, P. (2013). *Ontology Matching, Second Edition*. Springer.

Euzenat, J., Shvaiko, P., et al. (2007). *Ontology matching*, volume 18. Springer.

Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I. F., and Couto, F. M. (2013). The AgreementMakerLight Ontology Matching System. In *OTM Conferences - ODBASE*, pages 527–541.

Ferrara, A., Montanelli, S., Noessner, J., and Stuckenschmidt, H. (2011). Benchmarking matching applications on the semantic web. In *Extended Semantic Web Conference*, pages 108–122. Springer.

Galton, F. (1907). One Vote, One Value. *Nature*, 75:414.

Genta, L., Ferrara, A., and Montanelli, S. (2017). Consensus-based Techniques for Range-task Resolution in Crowdsourcing Systems. In *Proc. of the 7th EDBT Int. Workshop on Linked Web Data Management*, Venice, Italy.

Grau, B. C., Dragisic, Z., Eckert, K., Euzenat, J., Ferrara, A., Granada, R., Ivanova, V., Jiménez-Ruiz, E., Kempf, A. O., Lambrix, P., et al. (2013). Results of the ontology alignment evaluation initiative 2013. In *OM: Ontology Matching*, pages 61–100. No commercial editor.

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

Jiménez-Ruiz, E. and Cuenca Grau, B. (2011). LogMap: Logic-based and Scalable Ontology Matching. In *Int'l Sem. Web Conf. (ISWC)*, pages 273–288.

Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., and Berlanga, R. (2011). Logic-based assessment of the compatibility of UMLS ontology sources. *J. Biomed. Sem.*, 2.

Jiménez-Ruiz, E., Cuenca Grau, B., Zhou, Y., and Horrocks, I. (2012a). Large-scale interactive ontology matching: Algorithms and implementation. In *European Conf. on Artif. Intell. (ECAI)*, pages 444–449.

Jiménez-Ruiz, E., Grau, B. C., Horrocks, I., et al. (2012b). Exploiting the UMLS Metathesaurus in the Ontology Alignment Evaluation Initiative. In *2nd International Workshop on Exploiting Large Knowledge Repositories (E- LKR)*. CEUR- WS. org.

Li, H., Dragisic, Z., Faria, D., Ivanova, V., Jiménez-Ruiz, E., Lambrix, P., and Pesquita, C. (2019). User validation in ontology alignment: functional assessment and impact. *Knowledge Eng. Review*, 34:e15.

Malone, T. W., Laubacher, R., and Dellarocas, C. (2010). The Collective Intelligence Genome. *IEEE Engineering Management Review*, 38(3).

Mortensen, J. M. (2013). Crowdsourcing Ontology Verification. In *Proc. of the 12th Int. Semantic Web Conference*, pages 448–455, Sydney, Australia.

Ngomo, A.-C. N. and Auer, S. (2011). Limesa time-efficient approach for large-scale link discovery on the web of data. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Noronha, J., Hysen, E., Zhang, H., and Gajos, K. Z. (2011). Platemate: Crowdsourcing Nutritional Analysis from Food Photographs. In *Proc. of the 24th symposium on User Interface Software and Technology*, pages 1–12, Santa Barbara, CA, USA.

Noy, N. F., Mortensen, J., Musen, M. A., and Alexander, P. R. (2013). Mechanical Turk As an Ontology Engineer?: Using Microtasks As a Component of an Ontology-engineering Workflow. In *Proc. of the 5th ACM Web Science Conference*, pages 262–271, Paris, France.

Paulheim, H., Hertling, S., and Ritze, D. (2013). Towards Evaluating Interactive Ontology Matching Tools. In *Proc. of the 10th Extended Semantic Web Conference*, pages 31–45, Montpellier, France.

Röder, M., Saveta, T., Fundulaki, I., and Ngomo, A.-C. N. (2017). Hobbit link discovery benchmarks. In *Ontology Matching workshop at ISWC*.

Sarasua, C., Simperl, E., and Noy, N. F. (2012). CrowdMap: Crowdsourcing Ontology Alignment with Microtasks. In *Proc. of the 11th Int. Semantic Web Conference*, pages 525–541, Boston, MA, USA.

Saveta, T., Daskalaki, E., Flouris, G., Fundulaki, I., Herschel, M., and Ngonga Ngomo, A.-C. (2015). Pushing the limits of instance matching systems: A semantics-aware benchmark for linked data. In *Proceedings of the 24th International Conference on World Wide Web*, pages 105–106. ACM.

Thaler, S., Simperl, E. P. B., and Siorpaes, K. (2011). SpotTheLink: A Game for Ontology Alignment. In *Proc. of the 6th Conference on Professional Knowledge Management: From Knowledge to Action*, pages 246–253, Innsbruck, Austria.

Van Dusen, D. A., Chase, C., and Wise, J. A. (2016). System and Method for Fuzzy Concept Mapping, Voting Ontology Crowd Sourcing, and Technology Prediction. US Patent 9461876.

Volz, J., Bizer, C., Gaedke, M., and Kobilarov, G. (2009). Silk-a link discovery framework for the web of data. *LDOW*, 538:53.