A Framework to Evaluate the Quality of Integrated Datasets

Francesco Del Buono
University of Modena and Reggio Emilia
Modena, Italy
francesco.delbuono@unimore.it

Guglielmo Faggioli
University of Padua
Padua, Italy
faggioli@dei.unipd.it

Matteo Paganelli
University of Modena and Reggio Emilia
Modena, Italy
pagamatteo@gmail.com

Andrea Baraldi
University of Modena and Reggio Emilia
Modena, Italy
andrea.baraldi96@unimore.it

Francesco Guerra
University of Modena and Reggio Emilia
Modena, Italy
francesco.guerra@unimore.it

Nicola Ferro
University of Padua
Padua, Italy
ferro@dei.unipd.it

ABSTRACT

Evaluation is a bottleneck in data integration processes: it is performed by domain experts through manual onerous data inspections. This task is particularly heavy in real business scenarios, where the large amount of data makes checking all integrated tuples infeasible. Our idea is to address this issue by providing the experts with an unsupervised measure, based on word frequencies, which quantifies how much a dataset is representative of another dataset, giving an indication of how good is the integration process. The paper motivates and introduces the measure and provides extensive experimental evaluations, that show the effectiveness and the efficiency of the approach.

CCS Concepts

•Information systems → Mediators and data integration; Entity resolution; Deduplication;

Keywords

Entity Resolution, Entity Matching, Unsupervised Evaluation, Data Integration

1. INTRODUCTION

Data integration has always been considered as a key need for both research and industry. Traditionally the focus has been the integration of structured (typically relational) data sources where the information is divided into multiple tables. More recently, the attention paid to artificial intelligence and machine learning has led to the development of specific techniques for integrating datasets. From a technical perspective, these approaches typically implement a pipelined architecture, which consists of three major steps: schema alignment, entity resolution, and data fusion [16].

Despite the effort put by the research community (and partially reviewed in the Related Work Section), data and dataset integration is still far from being a solved problem and it is even less mature when applied in a real production context. Apart from the intrinsic complexity of the task, one of the barriers to fully empowering data integration is the human effort needed for evaluating and tuning the approaches. Indeed, you need to resort to controlled datasets, built on top of a manually created ground-truth, in order to compare your approach against this gold standard and score it accordingly. This is a long and economically demanding process, it presents serious challenges for scaling it up at the huge amounts of data needed in a real business scenario. Moreover, it is not able to keep pace with the quickly evolving data sources that you find in a real context and that call for a repeated over time and/or incremental integration process.

We address the problem from a completely different point of view, looking for an unsupervised way to measure how "good" is an integration process. By "good" we mean how much a data source is *representative* of another one, i.e. how much it preserves the informative content of another data source. Intuitively, the more a dataset can be represented by an integrated source, the less there is a loss of information when the integrated source is considered in place of the original one; we call this *input representativeness*. Viceversa, the more an integrated source can be represented by its datasets, the more it is consistent with them; we call this *output representativeness*.

Besides being an unsupervised measure, which reduces the required human effort and is suitable also for highly iterative and/or incremental real business scenarios, our approach considers the integration process as a whole and evaluates its quality after the data fusion step, which is what practitioners and domain experts are confronted with in a real context. On the contrary, most of the current literature [46, 64, 32] focuses on evaluating just the entity resolution step by using reference benchmarks and measures like precision or recall.

Finally, we introduce a technique to rank the elements from the input datasets and the integrated source according to their importance in the computation of the representativeness measures. The identification of the critical tuples, i.e., the tuples from the input datasets which are not adequately represented in the integrated source and the tuples from the integrated source which are redundant, allows the designer to improve the integration process.

This paper extends and consolidates the preliminary results introduced in [51, 50] by means of: (1) the introduction of an unsupervised metric for identifying the critical tuples for the integration process; (2) new experiments [1] based on shared datasets demonstrating the effectiveness and efficiency of the approach for high-dimensional and iterative / incremental integration processes; (3) an improved section of related work providing a deep review of the literature on the topic.

The paper is organized as follows: Section 2 presents our approach; Section 3 introduces some relevant scenarios and reports experiments about them; Section 4 discusses related works; finally, Section 5 draws conclusions and outlooks for future work.

2. THE APPROACH

2.1 Motivating Example

Data integration in real scenarios is usually performed via try and error approaches, requiring several iterations, where domain experts evaluate the correctness of the integrated datasets produced at each step. The integration strategy is improved and tuned at each step until the experts are satisfied with the result obtained.

Clearly, this is a fully manual and very demanding task in terms of time, effort, and resources required. We provide here an example of how this process works in practice to motivate the need for automatic and unsupervised tools for supporting it.

We use the popular "Cora Citation Matching" data[2] to create two datasets of pubblications – D_1 and D_2 shown in Table 1 – where each publication is described by a unique identifier, authors, title, and venue. Table 2 shows some possible results from their integration. In particular, Table 2a shows I_P, the perfect integration according to the Cora ground truth. On the other hand, Table 2b shows I_C, a low-quality integration, obtained by just concatenating entities for the two sources. As a result, some merges are missing from it, i.e. some items from D_1 and D_2 are not recognized as referring to the same entity; for example, publication *haussler1994* is mapped to two separate entities – respectively, the second and the last entity– instead of the same one. Finally, Table 2c shows I_M, another low-quality integration, obtained merging each entity in D_1 with an entity in D_2. Five entities in I_M are the result of a correct integration process, since they are also in I_P. The remaining 4 entities (which were not merged in I_P) are here randomly integrated. For example, the last entity, that refers to the publication *kearns1988b*, contains also information from the pubblication *kearns1994a*, which is therefore not recognized as a distinct entity.

A domain expert would manually assess the quality of I_C, and I_M, by: 1) randomly sampling (or based on "sentinel" elements defined a priori) a number of entities to check; 2) verifying their correctness; and, 3) categorizing erroneous outputs to support the development of improvements in the integration approach. An expert, analyzing the integrated dataset I_M, may discover that the second entity has been correctly created while the sixth one contains an error since it merges two items referring to different real world entities, i.e. *schapire1996* in D_1 and *schapire1997* in D_2. On the other hand, I_C contains two separate entries for the entity *kearns1990* which actually refer to the same entity and therefore are a duplication.

The effort required for performing the error analysis is very huge due to the large size of the datasets typically involved. An accurate evaluation requires scanning the entire integrated dataset searching for duplicated and/or wrongly merged entities and a comparison with the input datasets to verify that every real-world entity has been included in the final result. Moreover, since the integrated dataset is obtained after several try and error iterations, the error analysis is repeated multiple times. Therefore, an automatic tool for analyzing the quality of an integration process would largely reduce the effort required for performing an integration task.

2.2 The Model

We consider a dataset D as a collection of entities $D = \{e_1, \ldots, e_N\}$. The integration of datasets is performed by means of an entity integration function, defined below.

Definition 1 (Entity Integration process). *The Entity Integration process exploits an Entity Integration function (EI) to create an integrated dataset of entities $I = EI(\mathcal{D})$ from a collection of datasets $\mathcal{D} = \{D_1, \ldots, D_k\}$. The EI function defines the logic for matching and merging the entities in the input dataset collection $\mathcal{D}$.*

The integration approaches are usually evaluated with controlled datasets, pre-existing ground truths. Accuracy, and, more frequently, due to the unbalanced datasets, recall, precision, and F-measure are used to evaluate the quality of the integration result.

In business environments, the absence of a ground truth imposes to define a different procedure for the evaluation. The quality of the integration can be assessed through a *verification and validation process.* The verification process aims to check the formal correctness of the integrated dataset.

Definition 2 (Verified Entity Integrated Dataset). *The Entity Integrated Dataset $I = EI(\mathcal{D})$, where EI is an entity integration function applied to a collection of datasets $\mathcal{D} = \{D_1, \ldots, D_k\}$, should be:*

- ***total**: each entity of every input dataset should be represented in I, i.e., $\forall e_i \in D_k, \exists\ e_j \in I, s.t.\ e_j$ and e_i refer to the same real-world entity;*
- ***minimal**: I should not contain duplicated entities, i.e., $\forall e_i, e_j \in I, e_i$ and e_j refer to different real-world entities.*

The *validation process* assesses the correspondence of the informative content of the integrated dataset with the input sources.

[1] See the project github at `https://github.com/softlab-unimore/UEDI`

[2] `https://people.cs.umass.edu/~mccallum/data.html`

Table 1: Source datasets used in the motivating example.

(a) $\mathbf{D_1}$: the first data source.

entity id	authors	title	venue
freund1995a	yoav freund.	boosting a weak ...	in proceedings ...
haussler1994	haussler, d ...	rigorous learning ...	in proc. 7th ...
kearns1987	m. kearns, m. li ...	on the learnability ...	proceedings of ...
kearns1990	michael j. kearns.	the computational ...	
kearns1993b	m.j. kearns.	efficient noise-tolerant ...	in proc. 25th ...
schapire1996	r. e. schapire ...	learning sparse ...	j. of computer ...
kearns1994a	michael kearns, ...	on the learnability ...	proc. of the 26th ...
blum1994	avrim blum	weakly learning ...	in proceedings ...
freund1997a	yoav freund...	a decision-theoretic ...	journal of ...

(b) $\mathbf{D_2}$: the second data source.

entity id	authors	title	venue
freund1995a	freund, y.	boosting a weak ...	in 'proceedings ...
haussler1994	haussler ...	rigorous learning ...	in proceedings ...
kearns1987	m. kearns ...	on the learn-ability ...	in proc. 19th stoc,
kearns1990	michael ...	the computational ...	
kearns1993b	m. kearns.	efficient noise-tolerant ...	in proceedings ...
haussler1994a	d. haussler, ...	bounds on the sample ...	machine learning,
kearns1988b	michael kearns.	thoughts on ...	(unpublished),
schapire1997	schapire, r.e ...	w.s.: boosting ...	proceedings of ...
rivest1989	r. l. rivest ...	inference of ...	in acm symposium ...

Table 2: Three possible integrated datasets.

(a) $\mathbf{I_P}$: the *P*erfect integrated dataset.

id	entity id	...
1	*freund1995a*	...
2	*haussler1994*	...
3	*kearns1987*	...
4	*kearns1990*	...
5	*kearns1993b*	...
6	*schapire1996*	...
7	*schapire1997*	...
8	*blum1994*	...
9	*freund1997a*	...
10	*haussler1994a*	...
11	*kearns1988b*	...
12	*kearns1994a*	...
13	*rivest1989*	...

(b) $\mathbf{I_C}$: low quality integrated dataset (concatenation).

id	entity id	...
1	*freund1995a*	...
2	*haussler1994*	...
3	*kearns1987*	...
4	*kearns1990*	...
5	*kearns1993b*	...
6	*schapire1996*	...
7	*schapire1997*	...
8	*blum1994*	...
9	*freund1997a*	...
10	*haussler1994a*	...
11	*kearns1988b*	...
12	*kearns1994a*	...
13	*rivest1989*	...
14	*kearns1987*	...
15	*kearns1990*	...
16	*kearns1993b*	...
17	*freund1995a*	...
18	*haussler1994*	...

(c) $\mathbf{I_M}$: low quality integrated dataset (merging).

id	entity id	...
1	*freund1995a*	...
2	*haussler1994*	...
3	*kearns1987*	...
4	*kearns1990*	...
5	*kearns1993b*	...
6	*schapire1996,* *schapire1997*	...
7	*blum1994,* *rivest1989*	...
8	*haussler1994a,* *freund1997a*	...
9	*kearns1988b,* *kearns1994a*	...

The unsupervided technique for evaluating EI processes proposed in this paper is based on a *representativeness function* that scores how much a dataset D_1 can be represented by a second dataset D_2 through the loss of information in using D_2 instead of D_1. We decided to implement the *representativeness function* by analyzing the word frequency distribution in the datasets.

Definition 3 (Word frequency distribution in datasets). *Given a dataset D, let V be its vocabulary of terms. The word frequency distribution $freq_D(w) : V \rightarrow \mathbb{N}_0$ of the dataset D is a function which associates each term $w \in V$ with its frequency in D.*

The simplest approach for the definition of a vocabulary of terms V for a dataset is to apply a tokenization algorithm to the concatenation of all tuples in D. Token splitting can be considered as a solved problem [65] and a large number of techniques are available in NLP code libraries.

Definition 4 (Dataset representativeness score). *Given two datasets D_1 and D_2, the dataset representativeness $r_{D_1 \rightarrow D_2}$ quantifies the extent to which dataset D_1 represents D_2 by measuring how much the word frequency distribution $freq_{D_1}$ approximates $freq_{D_2}$.*

In the next section, we propose a way to measure the approximation between two word frequency distributions in the context of a data integration process. The representativeness score should provide users with an assessment of how much datasets are represented by integrated sources by showing if there is any loss of information; vice-versa, it should quantify how much integrated sources are represented by the original datasets by showing if there is any redundancy or irrelevant content.

2.3 Scoring Representativeness

When assessing the quality of the integration process, we need to consider the two sides of the coin, i.e. how well a source D is represented by the integration I and, vice-versa, how well the integration I is represented by a source D.

If the integration process is perfect, we expect that the content of D is completely "covered" by the content of I. This means that the vocabulary used in D should be included in the vocabulary used in I, and the word frequency distribution of words in D should be less than or equal to the one in I. The measure of the coverage of these word frequency distributions can provide a measure of the representativity of an integration source for a dataset. We call this measure *input representativeness* $r_{D \rightarrow I}$ and we define it in Equation (1).

Definition 5 (Input representativeness). *Given two datasets D and I, where I is the integration of D according to some EI function, let V_D be the vocabulary of D and $freq_X(w)$ be the word frequency distribution of either D or I. We define the following representativeness score:*

$$r_{D \rightarrow I} = 1 - \frac{1}{|V_D|} \cdot \sum_{w \in V_D} \frac{freq_D(w) - min(freq_D(w), freq_I(w))}{max(freq_D(w), freq_I(w))} \quad (1)$$

We expect that an integrated dataset contains more entities than an input dataset, due to the contribution of other

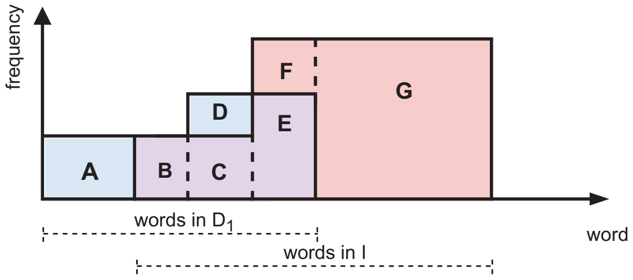


Figure 1: Example of word distributions.

datasets. Nevertheless, excluding stop words and other very generic words, we can suppose that the distribution of frequencies of words belonging to the intersection of the vocabularies of I and D is close. By measuring this closeness, we can evaluate how much the dataset can represent its integration for the shared words. We call this measure *output representativeness*, $r_{I\rightarrow D}$, and it is defined in equation (2).

Definition 6 (Output representativeness). *Given two datasets D and I, where I is the integration of D according to some EI function, let V_D be the vocabulary of D and $freq_X(w)$ be the word frequency distribution of either D or I. We define the following representativeness score:*

$$r_{I\rightarrow D} = 1 - \frac{1}{|V_D|} \cdot \sum_{w \in V_D} \frac{freq_I(w) - min(freq_D(w), freq_I(w))}{max(freq_D(w), freq_I(w))} \qquad (2)$$

We observe as the output representativeness $r_{I\rightarrow D}$ is defined over the vocabulary V_D of the dataset D and not on the vocabulary of the integration I. Indeed, there is an intrinsic asymmetry in the integration process and we need to keep the focus on the dataset D, either considering how much it is represented by the integration I, i.e. $r_{D\rightarrow I}$, or how much it represents the integration I, i.e. $r_{I\rightarrow D}$, but without skewing the scores by including all the terms of V_I. Considering the whole vocabulary V_I, and not just its overlap with V_D, would just bring in all the other sources than D, whose vocabulary may differ a lot from V_D, and, as a result, these additional (and possibly unrelated) terms would mask how much D and I represent each other.

Example 1. *Figure 1 shows a simplified word frequency distribution for a dataset D_1 and its integration I. The x-axis represents the words found in the data sources and the y-axis their respective distribution. Note that, for sake of simplicity, the heights of the frequency histograms are approximated to three possible values and the actual words are not reported on the x-axis. The areas A, B, C, D, E represent the word frequency distribution for D_1 and the areas B, C, E, F, G the one of I. A and G represent words belonging only to the input dataset and integrated dataset respectively. The words in B, C, D, E, F are common to both the sources and: (1) those of B have the same frequency distribution; (2) those of C and D have frequency distribution equal to C in the integration and frequency distribution equal to $C + D$ in the input dataset; (3) those of E and F have frequency distribution equal to E in the input dataset and frequency distribution equal to $E + F$ in the integration. To have a high value of representativeness, (1) the frequency of the common terms in the datasets should be similar (i.e. the regions D and F have to be as small as possible), and (2) a small number of terms should be used in a dataset only (i.e. the area of region A is limited). This is the behavior modeled by equations 1 and 2, which correspond to $r_{D\rightarrow I} \propto 1 - (A + \frac{D}{C+D})$, and $r_{I\rightarrow D} \propto 1 - (\frac{F}{E+F})$ when applied to the scenario represented in Figure 1.*

2.4 Representativeness Supporting the Verification

The representativeness score can be used to *verify* an integration process, where the *input representativeness* score measures the *totality* of the integrated dataset; the *output representativeness* score the *minimality* of the integrated dataset.

Let I be obtained by the integration of D_1 and D_2. The input representativeness of I with respect to the input datasets D_1 and D_2 is obtained by averaging their input representativeness scores (i.e. $r_{D_1\rightarrow I}$ and $r_{D_2\rightarrow I}$). This aggregated score provides a measure of the totality of the integration process, since the more I represents the sources D_1 and D_2, the more the entities of D_1 and D_2 are also in I. On the other side, the output representativeness of D_1 and D_2 with respect to I, obtained by averaging $r_{I\rightarrow D_1}$ and $r_{I\rightarrow D_2}$, is a measure of the minimality of the integration process. Indeed, if D_1 and D_2 have high output representativeness, it follows that I does not contain duplicated entities.

2.5 Representativeness Supporting the Validation

An integration process can be *validated* by plotting the representativeness scores in a two-dimensional Cartesian plane. The x-axis reports the *input representativeness* $r_{D\rightarrow I}$, i.e. the *totality*, and shows the values obtained by the datasets with respect to the integration; the y-axis reports the *output representativeness* $r_{I\rightarrow D}$, i.e. the *minimality*, and shows the behavior of the integration with respect to the input sources. Values closest to the point $(1, 1)$ represent the best performance. We call the distance from (1, 1) *representativeness distance* and we claim that this is a measure of the validation of an integration approach. Indeed, the more we depart from (1, 1), the more the correspondences between entities in the input and integrated datasets decreases. Note that only in ideal scenarios, where the entities are represented in the input datasets with the same property values, the combined representativeness score of a verified and validated integrated dataset is (1, 1). Often, data representing the same entities are not the same, due to updates, mismatches and mistakes. This affects the word frequency distributions of the corresponding datasets which will have small differences and make representativeness values departing from (1,1).

Example 2. *Figure 2 shows the values of the representativeness scores obtained for the I_P, I_C, and I_M integrated datasets, described in Section 2.1. As expected, I_P is the best integrated dataset, being the closest to the point (1,1). We observe that I_C is the integration that better represents the input datasets since it has the highest values for the in-*

Scenario	input repr.	output repr.
$D_1 \rightarrow I_C$	1	0.626
$D_2 \rightarrow I_C$	1	0.538
$D_1 \rightarrow I_M$	0.712	0.953
$D_2 \rightarrow I_M$	0.633	0.918
$D_1 \rightarrow I_P$	1	0.889
$D_2 \rightarrow I_P$	0.917	0.849

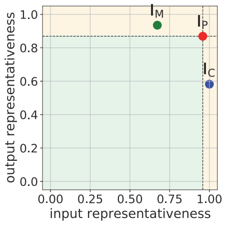


Figure 2: Input and output representativeness for the sources of the motivating example.

put representativeness. *It is the concatenation of the input datasets, so the resulting input representativeness value is 1, since the input word frequency distribution is completely included in the integrated dataset. Nevertheless, I_C obtains the worst value of* output representativeness*, thus meaning that it contains duplicated entries. I_M shows the highest results for the* output representativeness*. I_M has been built minimizing duplicated items (all entries in the input datasets have been merged). The worst values obtained for the* input representativeness *score means that the integrated dataset does not completely represent the input datasets. This is due to the wrong entity-merges that we have introduced. Note that input and output representativeness have to be jointly evaluated and the values assumed by the ground truth (I_P in the example) do not constitute an upper bound for the values that input and output representativeness can assume. In Figure 2, I_M and I_C are both located in the yellow area, which includes the elements with representativeness value greater than the one of the ground truth for at least one dimension. Nevertheless, even if I_M has a higher value of output representativeness, the quality of I_M (as the distance from (1,1) shows) is worst than the one of I_P due to the lower input representativeness. The same happens for the quality of I_C, which is worst than the one of I_P due to the lower output representativeness.*

2.6 Ranking Tuples According to Representativeness

Being able to identify and analyze the tuples that generate mistakes in the integration process is helpful to improve the next iterations of the process itself. In particular, tuples from the datasets which are not represented in the integrated source affect the input representativeness score. To be able to recognize these tuples allows the integration process developers to design broader functions customized for including the missing tuples in the integration. On the other hand, duplicated entries in the integrated dataset decrease the output representativeness. Making the developers aware of them allows the development of more specialized integration functions able to recognize duplicated information. We address this issue by introducing two measures, which are applied to the input datasets and the integrated source, allowing the ranking of the tuples according to their contribution to the input and output representativeness, respectively.

Table 3: The use cases considered. T = Textual, D = Dirty, and S = Structured dataset.

Use Case	Name	Input Datasets	Integrated Dataset	Shared Entities (%)	Unique Entities (%)
U1	T Abt-Buy	$\|D_1\| = 949 - \|D_2\| = 920$	$\|I\| = 1174$	58.5	41.5
U2	S Amazon-Google	$\|D_1\| = 1171 - \|D_2\| = 1843$	$\|I\| = 2232$	32.7	67.3
U3	S Beer	$\|D_1\| = 237 - \|D_2\| = 233$	$\|I\| = 412$	14.1	85.9
U4	S Fodors-Zagats	$\|D_1\| = 89 - \|D_2\| = 238$	$\|I\| = 422$	24.9	75.1
U5	D iTunes-Amazon	$\|D_1\| = 272 - \|D_2\| = 278$	$\|I\| = 450$	20.9	79.1
U6	S iTunes-Amazon	$\|D_1\| = 251 - \|D_2\| = 255$	$\|I\| = 410$	22.2	77.8
U7	D DBLP-ACM	$\|D_1\| = 2419 - \|D_2\| = 2238$	$\|I\| = 2511$	85.5	14.5
U8	S DBLP-ACM	$\|D_1\| = 2406 - \|D_2\| = 2220$	$\|I\| = 2507$	84.5	15.5
U9	D DBLP-GoogleScholar	$\|D_1\| = 2491 - \|D_2\| = 9877$	$\|I\| = 7959$	29.0	71.0
U10	S DBLP-GoogleScholar	$\|D_1\| = 2488 - \|D_2\| = 9286$	$\|I\| = 7865$	29.0	71.0
U11	D Walmart-Amazon	$\|D_1\| = 1578 - \|D_2\| = 4297$	$\|I\| = 5080$	14.0	86.0
U12	S Walmart-Amazon	$\|D_1\| = 1524 - \|D_2\| = 4014$	$\|I\| = 4784$	14.0	86.0

$$rank_{D\rightarrow I}(e) = 1 - \frac{1}{|V_e|} \cdot \sum_{w \in V_e} \frac{freq_D(w) - min(freq_D(w), freq_I(w))}{max(freq_D(w), freq_I(w))} \tag{3}$$

Equation 3, specializing Equation 1 to evaluate tuples, provides the measure of how much the words in a tuple are represented in the integrated source. Its application to all tuples of the input datasets allows to identify the elements which mostly contribute in decreasing the input representativeness score (the ones with the score closest to 0).

$$rank_{I\rightarrow\mathcal{D}}(e) = \max_{\forall D \in \mathcal{D}} (1 - \frac{1}{|V_e \cap V_D|} \cdot \sum_{w \in V_e \cap V_D} \frac{freq_I(w) - min(freq_D(w), freq_I(w))}{max(freq_D(w), freq_I(w))}) \tag{4}$$

Equation 4, specializing Equation 2 to evaluate tuples from the integrated source, provides the measure of the extent to which the words of the integrated source cover the input datasets. Its application to the tuples of the integrated source allows to identify the elements which mostly contribute in decreasing the output representativeness score.

3. EXPERIMENTAL EVALUATION

We conduct a quantitative (in Sections 3.2 to 3.4) and qualitative (in Section 3.5) evaluation of the effectiveness of our proposed measures. Finally, in Section 3.6, we assess their efficiency.

3.1 Experimental Setup

We use 12 publicly available use cases (see Table 3) from the benchmark of the Magellan tool[3], that is the main reference to evaluate entity matching approaches. The use cases consist each one of two datasets of entities and the ground truth contains pairs of entities, one for each dataset, labelled as matching and non matching items. According to the literature [21], we consider entities as referring to the same real world entity when the matching elements form a clique. In this case, we adopt a simple merging strategy by randomly

[3] `https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md`

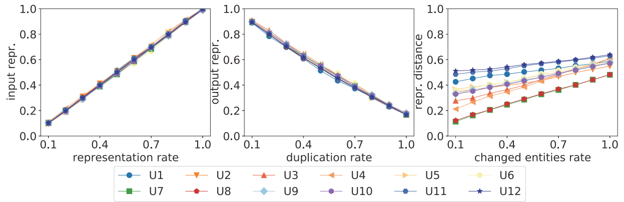


Figure 3: Verification and validation: measures.

selecting one of the entity as the one resulting from the merging process. The third and fourth columns in Table 3 show the cardinalities of the input and integrated datasets. The Table also shows for each use case the ratio of shared entities (i.e., entities in the integrated dataset which are generated by merging more input entities) and unique entities (i.e., entities which come from one of the input sources only). The distribution of these kinds of entities in the ground truth is typically unbalanced: only in U1 shared and unique entities have a similar distribution.

We run all experiments on commodity hardware: a server with 4 virtual cores, 16GB of RAM, 256GB of local (SSD) storage and that runs Ubuntu version 20.04.

3.2 Verification and Validation of an Integration Process

We evaluate the extent to which representativeness supports the verification and validation of an integration process: input representativeness for the totality, output representativeness for the minimality, and representativeness distance for the validation. The idea of the experiment is to modify the datasets in a controlled way and to check if the representativeness measures vary as expected to reflect the changes.

The first plot on the left of Figure 3 shows how the input representativeness scores measures the totality of the integration process. For each use case, a number of integrated datasets have been created from the ground truth, by selecting an increasing percentage of ground truth entities, as specified in the x-axis. The input representativeness score computed with these reduced datasets is shown on the y-axis. We expect that low input representativeness scores correspond to integrated datasets composed of a reduced numbers of entities. This is due to the existence of entities in the input datasets that do not have any correspondence in the integration. The first plot in Figure 3 shows that the score increases with the number of entities included in the integrated dataset. In a similar way, the second plot shows on the x-axis the percentage of duplicated entities that we have introduced in a ”perfect” integrated dataset and, on the y-axis, the corresponding output representativeness score. As expected, the higher the number of duplicates, the lower the value of the output representativeness. Finally, the third plot on the right of Figure 3 evaluates how well the representativeness distance measures the validation of an integration approach. We alter the datasets by removing and by duplicating the same percentage of entities; therefore, for example, a value of 10% on the x-axis means that 5% of the entities are duplicated and 5% are removed; the y-axis shows the corresponding value of the representativeness distance. As expected, the distance grows with the increase of duplicated and missing entities, providing an overall validation of the process. Note that the slope of the curves is less sharp than the previous ones. This is due to the joint contribution of the input and output representativeness in the definition of this measure. Indeed, an entity duplication generates both a reduction of the minimality and an increase of the totality.

Take-away: the input and output representativeness are effective implementations of the totality and minimality properties respectively, while the representativeness distance is a valuable validation measure for an integration process.

3.3 Quality of the Representativeness Scores

3.3.1 Robustness to Randomness in the Data

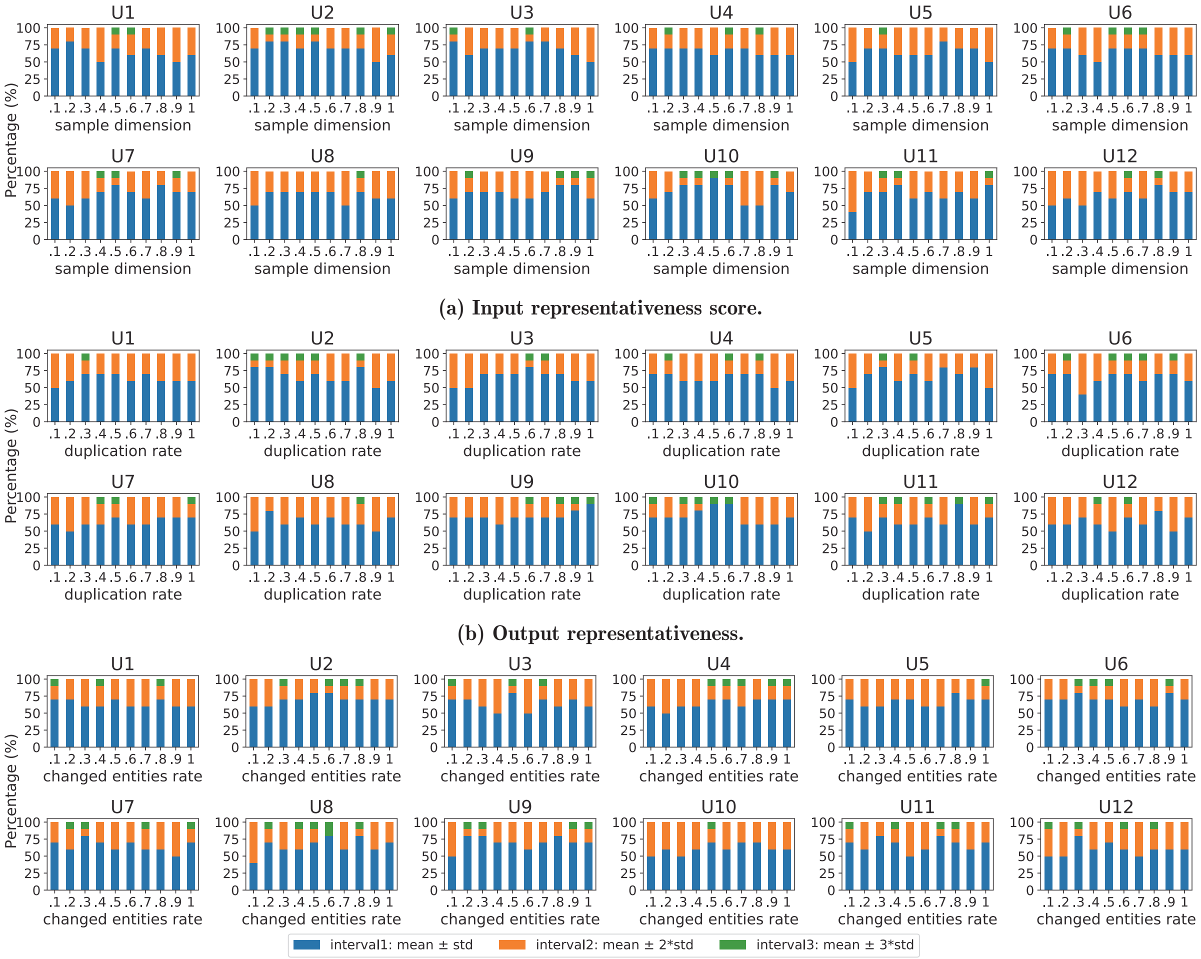
We assess to what extent randomness affects our proposed representativeness scores. To this end, for each representativeness score, we repeat 100 times each of the three experiments reported in the previous Section 3.2 by randomly and uniformly sampling with replacement the data used in each configuration of the experiment. In this way, we can compute mean and standard deviations for each score (i.e., the input, output, and distance representativeness) and verify how often a given score falls in the expected range as defined in Figure 3. Indeed, the more a score falls in the expected range using random and equivalent samples of the same data, the more robust is its predictions, and the less we would change our conclusions due to the observed sample.

Figure 4 shows the results of this experiment for each representativeness score and use case. We considered three ranges: one standard deviation in blue; two standard deviations in orange; and, three standard deviations in green. Each bar in the histograms indicates which ratio of the 100 scores falls in the blue, orange, or green interval. For example, in Figure 4a for use case U1 and a deterioration of 50% of the samples, i.e. 50% of the entities have been removed in this case, we can observe that roughly 70% of the input representation scores fall in the one standard deviation range (blue bar); 20% in the two standard deviations range (orange bar on top of the blue one); 10% (or less) in the three standard deviations range (tiny green bar on top of the orange one).

In the case of the input representativeness in Figure 4a we can observe as the scores fall in the one standard deviation range in 50% to 75% of the cases, indicating a quite stable measure; almost all the other cases fall in the two standard deviations range, and just few of them in the three standard deviations range. We can observe a similar behaviour also for the output representativeness in Figure 4b and for the representativeness distance in Figure 4c.

3.3.2 Robustness of the Representativeness Scores Varying the Dataset Size

In this experiment we evaluate if the representativeness measures vary as the size of the considered datasets varies. The more the measures are stable, the more the approach is robust to the randomness of the data in the datasets. This experiment provides a complementary assessment compared to previous experiments that focused on the variability of



(a) **Input representativeness score.**

(b) **Output representativeness.**

(c) **Representativeness distance.**

Figure 4: Ratio of representativeness scores.

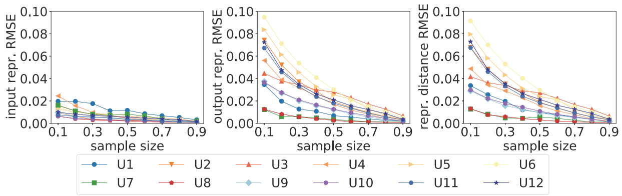


Figure 5: RMSE between the mean representatives scores and the ideal score based on the ground truth.

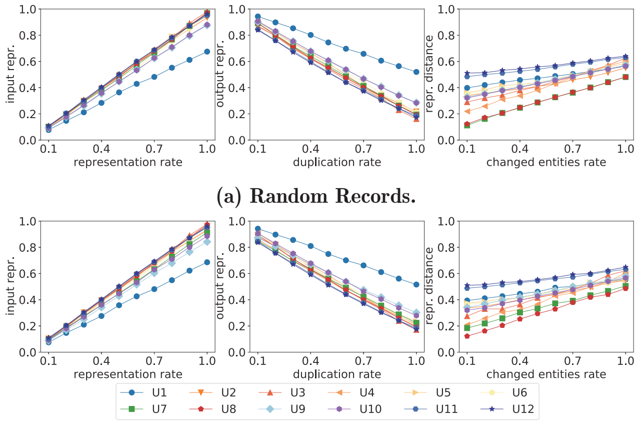
results. We selected samples of increasing size from the ground truth (equal to 10%, 20%, ..., 100%) and we repeated this sampling process 100 times for each target size. For each type of representativeness score, Figure 5 shows the *Root Mean Square Error* (RMSE) between the score computed using the entire ground truth and the mean score computed over the samples related to a target sample dimension. The representativeness metrics do not show significant variations: only for use cases containing small datasets there are higher variations, although never greater than 0.1. This demonstrates their robustness even when significant changes in the size of the involved datasets are applied.

3.3.3 *Robustness to the Selected Merging Approach*

We evaluate how much the behaviour of our representativeness measures depends on the actual merging strategy adopted to perform the integration of the matching entities. Ideally, we would like to observe some differences in the scores but not drastically different behaviours, otherwise, we could not reliably compare alternative integration processes. To this end, we repeat the experiment of Section 3.2 but we use two different alternatives for merging. Figure 6a shows the results for the first approach which randomly selects which entities to merge. Figure 6b shows the results for the second approach which randomly selects the values of the merged attributes. In both figures, we can observe a trend which is consistent with all the previous experiments.

3.3.4 *Robusteness of the Ranking*

To evaluate if Equations 3 and 4 can really detect the mis-



(a) Random Records.

(b) Random Values.

Figure 6: Verification and validation of the representativeness measures against random merging strategies.

taken tuples, we build five integrated datasets for each use case where we have modified 10%, 20%, 30%, 40%, and 50% of tuples to introduce mistakes. In particular, we removed tuples to evaluate the input representativeness, and we inserted duplication to evaluate the output representativeness. Then we applied Equations 3 and 4 and we evaluated the percentage of really mistaken tuples in the first ten highest-scored tuples (precision at 10, p@10), in the first thirty (p@30) in the first fifty (p@ 50), in the first eighty (p@80), and in the first one hundred (p@100). Figure 7 shows the results of our experiment, where missing columns denote the absence of enough errors to perform the computation. In most of the configurations, our measures demonstrate high precision levels making them really able to detect mistakes.

Take-away: the proposed representativeness scores are quite robust to different types of deterioration and randomness in the data, have a good predictive accuracy, and they are not biased by the considered data fusion techniques. Moreover, simple variations of the scores as the ones in Equations 3 and 4 allows the detection of mistaken tuples, thus making the developer able to design improvements in the integration process.

3.4 Alternative Techniques for Measuring Input Representativeness

In Section 2.3 we proposed a specific way of computing representativeness based on word frequency distributions computed on the whole dataset. These distributions can be inaccurate for describing entity similarities, computed at the tuple level.

In this section, we consider the following alternatives for computing the input representativeness score: the jaccard-based similarity as a baseline, for its simplicity; the bleu-score [57] as a reliable unsupervised measure for evaluating the quality of machine-translated text; finally, embeddings largely used in NLP tasks to capture both syntactic and semantic similiarity.

Jaccard and Bleu score-based representativeness. Firstly, we tokenize the entries in the input and the integrated datasets and then we measure the similarity between input and integrated entities. For each input entity, we consider the maximum value computed. The mean of all maximum values is the representativeness measure for the considered input data source.

Embedding-based representativeness. We applied three different techniques (word2vec [45], fasttext [7], and glove [59]) for computing the embeddings of the tokenized entries of input and integrated entities. We measured the similarity between input and integrated entities through the cosine similarity. For each input entity, we consider the maximum value computed and we average the results for all the entities as before.

3.4.1 *Alternatives for the Representativeness Distance*

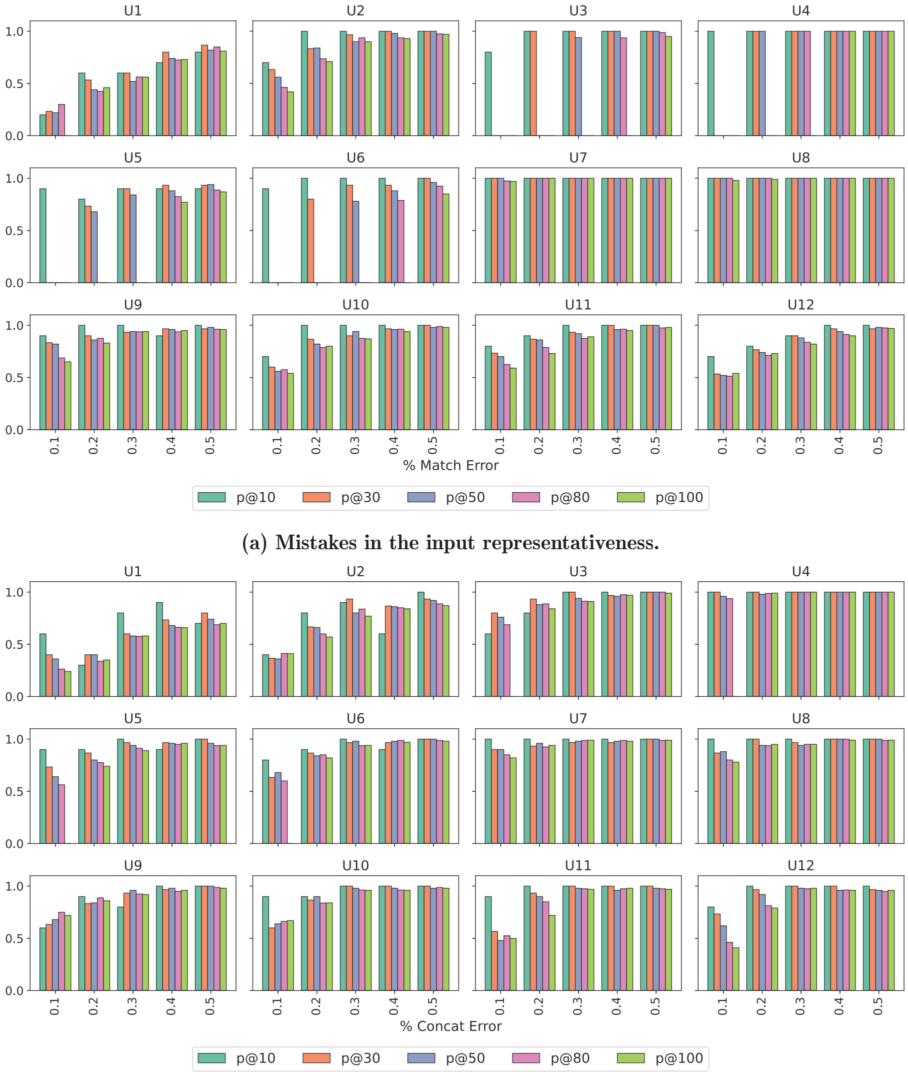
We conduct the experiment described in Section 3.2 comparing the representativeness distance obtained with the alternative measures. Figure 8 shows the results obtained. As expected, the representativeness distance increases as the deterioration of the datasets increases. Nevertheless, the measure defined in Equation 1 assumes the highest values in the majority of the scenarios, indicating that it better recognizes the errors in the integrated dataset.

Take-away: our measure outperforms alternative representativeness metrics based on syntactic and semantic similarities.

3.4.2 *Alternatives for Input and Output Representativeness*

We compare alternative representativeness measures on the basis of how they react to possible errors in the integration process. We consider two error types: items in the input dataset which are merged even if they represent different entities and items referring to the same real world entities which are not merged. Note that this experiment may resemble the one of Section 3.2 but here we operate directly on the input datasets and on the different categories of entities.

Let us consider the "merge errors". We defined as *unique entities* those entities in the input datasets which are not to be merged with other entities in the integration process. When we erroneously merge unique entities with other entities, the dimension of the integrated dataset decreases as well as its totality, since there are input entities which are not represented in the integrated dataset, i.e. the wrongly merged ones. As a consequence, this kind of error will affect the input representativeness. To evaluate the impact of these errors, we created variations of the use case datasets, where different amounts of errors have been introduced in the ground truth, and we measured the difference of the input representativeness score measured with respect to the ground truth. The results of the experiments are shown in Figure 9, where for each use case, selected percentages of wrong merged entities have been introduced. The input representativeness (independently from the approach used for its computation) decreases when the error increases in all use cases and with all the approaches. Nevertheless, we observe that our measure introduced in Equation 1 better represents these mistakes, by showing the largest variations.



(a) **Mistakes in the input representativeness.**

(b) **Mistakes in the output representativeness.**

Figure 7: Evaluation of the equations for finding mistakes.

Note that Figure 9 shows the results on the overall dataset, not only on the portion of the dataset composed of unique entities. The unbalanced distribution of unique entities (see Table 3) can introduce different amounts of wrong merges in the use cases. Table 4 shows the “real” impact of the perturbations introduced in the ground truth, by showing the percentage of missing unique entities for each experiment. We see that the variation in use cases U7 and U8 are less marked since the reduced number of wrong entities introduced. The plots describing U3, U11, and U12 are those with the largest variations, and this is consistent with the perturbed integrated entities.

Table 4: Percentage of unique entities removed from the integrated dataset for each experiment.

	U1	**U2**	**U3**	**U4**	**U5**	**U6**	**U7**	**U8**	**U9**	**U10**	**U11**	**U12**
0.2	8.26	13.44	17.23	15.17	16	15.61	2.91	3.07	14.19	14.16	17.22	17.18
0.4	16.52	26.88	34.47	30.09	31.56	30.98	5.81	6.18	28.4	28.35	34.41	34.36
0.6	24.96	40.41	51.46	45.02	47.56	46.83	8.72	9.29	42.58	42.52	51.63	51.55
0.8	33.22	53.85	68.69	59.95	63.11	62.2	11.63	12.41	56.79	56.71	68.82	68.73
1	41.48	67.29	85.92	75.12	79.11	77.8	14.54	15.48	70.98	70.87	86.04	85.91

We conduct a similar analysis for the second issue, i.e. duplicated entities. We called *shared entities* those entities obtained from merging multiple input entities. In this case, errors in the shared entities result on items in the integrated dataset which are not merged and this will affect the output representativeness. As before, we create a controlled deterioration of the ground truth, where we introduce errors on 20%, 40%, ...100% of the shared entities. Figure 10a shows the difference of the output representativeness score with respect to the ground truth: the more the decrease, the more errors in shared entities are detected. Note that, as before, Figure 10b is needed to support the analysis. It shows the percentage of new entities introduced with the

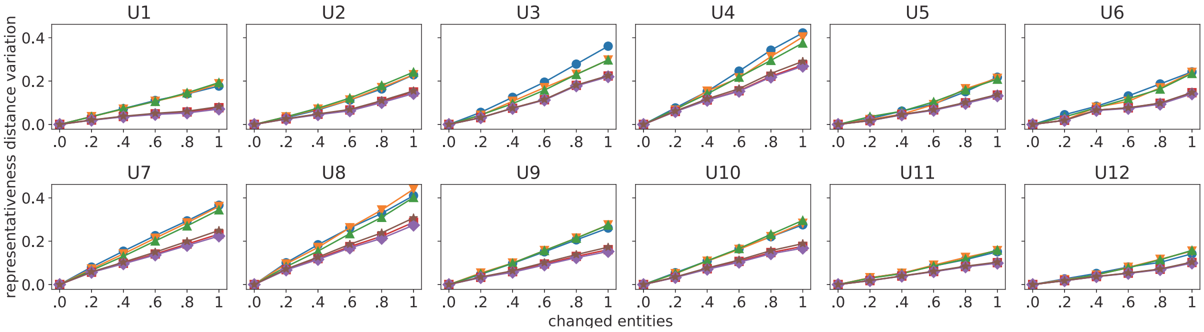


Figure 8: Comparison among the measures introduced for computing the representativeness distance.

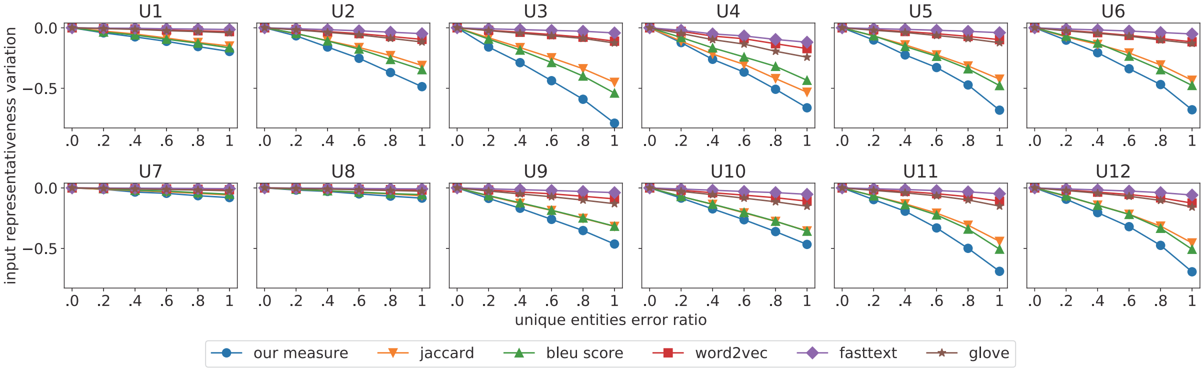
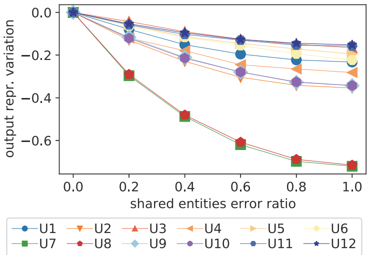


Figure 9: Impact of wrongly merged entities on the input representativeness.

perturbation: U7 and U8 show the largest amount of entity introduced. This is consistent with the results in the figure that show the largest variation.



(a) Output representativeness variation in case of non-merged entities.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12
0.2	21.55	12.46	4.13	8.77	7.33	7.07	30.63	30.55	16.15	15.04	5.45	5.33
0.4	37.56	22.49	8.01	14.69	12.67	11.46	54.4	53.45	30.66	28.2	9.78	9.62
0.6	49.66	29.48	10.92	20.85	16.89	16.1	71.64	70.68	42.73	38.88	12.93	12.86
0.8	56.64	33.29	12.62	22.99	19.78	20.24	82.04	80.93	51.84	46.68	15.02	15.07
1	59.2	35.04	14.08	24.88	22.22	23.41	85.46	84.52	55.4	49.7	15.65	15.76

(b) Percentage of duplicated entities introduced in the integrated dataset for each use case.

Figure 10: Impact of errors on shared entities on the input representativeness.

3.4.3 *Joint Effect of Duplication and Merging Errors*

We analyze variations of both the input and output representativeness scores resulting from the increase of wrong merged entities and wrong duplicated entities in the datasets. Figures 11 and 12 show the results of our experiments. Green arrows show the variations on the primary component (input representativeness in the first case and output representativeness in the second one). Red arrows show the secondary component. The longer the arrow, the higher the variation in the score. The scores range from -1 to 1. They represent the difference between the value assumed by the representativeness measured in the experiment and the one in the ground truth.

In Figure 11, green arrows are associated with the input representativeness, red to the output representativeness. The perturbations of the datasets are generated by introducing wrong merged entities. As already shown in Figure 11, the input representativeness decreases with the increase of the errors. This is shown by the green arrows which become longer and tend to -1 in correspondence of the largest perturbations. We observe that the red arrows have an opposite behavior: they increase when the wrong merged entities increase. This is due to the fact that increasing the number of wrong merged entities increases the minimality of the integrated dataset.

In Figure 12 we show the results of the experiments with datasets where duplicated entities have been introduced. In this case, the green arrows show the output representativeness, which is the measure subject to the largest variations. As in Figure 10, in correspondence of the largest perturbations, the values assumed by the output representativeness are closer to -1. In this situation, even if less marked, we can observe an increase of the input representativeness. This is due to the fact that an increase of duplicated entities in the integrated dataset increases also its ability to represent the input sources.

Take-away: by examining the variations of input and output representativeness, we understand the error affecting the integration task. A main variation of the input representativeness indicates errors in the recognition of the no-match class. Errors in the match class produce a more marked variation in the output representativeness.

3.5 Controlled Data Integration Scenarios

Creating the datasets. For each use case in Table 3 we generate four datasets, D_1, D_2, D_3 and D_4. D_1 has a cardinality double than D_2 which has the same cardinality as D_3. D_2 contains a subset of the entities of D_1. D_3 contains entities that are not in D_1. D_4 concatenates D_2 and D_3. We evaluate the datasets in three controlled scenarios. The first column in Table 5 shows the cardinalities of the datasets and the associate vocabularies. The datasets are experimented in three controlled scenarios.

Scenario 1: Datasets describing the same entities. We consider D_1 and D_2, which describe same entities. Since D_1 is a superset of D_2, it can be considered as a possible integration, called $I_M = D_1$ in Figure 13a. I_C is the integration obtained by a concatenation of the tuples in D_1 and D_2. Let us consider for example use case U10: we know the ground-truth and it is thus possible to compute the error rate, which is 0 for I_M, and 0.333 for I_C. Our measure shows that the concatenation I_C is the best integration scenario, since it does not generate loss of information. This is clear in Figure 13a, where I_C assumes the maximum value of input representativeness on the x-axis. Nevertheless, concatenation introduces data duplication (D_1 is a superset of D_2) and this is the reason why in Figure I_C has an output representativeness value on the y-axis lower than I_M. The plot clearly shows that I_M is a better integration than I_C, as we can expect by analyzing the data sources.

Scenario 2: Datasets describing different entities. We consider D_1 and D_3, which describe different entities. As in the previous scenario, we consider D_1 also as integration and we call it I_M in Figure 13b. I_C is the integration obtained by the concatenation of D_1 and D_3, which does not contain duplicates in this case. In this scenario, I_C should be the best integration since all entities are included in this source. This is confirmed by the error rate, 0.5 for I_M and 0 for I_C. This is also clear by our measure applied to U10 (see Figure 13b), comparing the coordinates of I_C and I_M in the Figure. I_C has coordinates (1, 0.79). This means the maximum input representativeness value. I_M has coordinates (0.73,0.9). The output value is due to the low representativeness value for D_3 in I_M (0.46). Note that even if I_M does not contain the entities described in D_3 the representativeness is not zero since there is still a low number of words in D_3 which are contained in I_M anyway. The high level measured from the integration perspective is because I_M completely includes D_1 which has twice the cardinality of D_3.

Scenario 3: Datasets describing common entities. We consider D_1 and D_4 which contain a half common and a half different entities. I_P in Figure 13c is generated by concatenating D_1 and D_3. This is a perfect integration since it includes all entities described by the D_1 and D_4 datasets. I_M, as in the previous scenarios, is D_1 only which, in this case, does not describe half of the entities in D_4. Finally, I_C is obtained by the concatenation of D_1 and D_4. This integration suffers from redundancy, generated by the duplicated entities of D_1 contained in D_4 and included twice in I_C. The error rates of these integrations are 0.5 for I_M and I_C, and no error rate for I_P. Figure 13c shows our measures applied to U10 and correctly reflects the datasets included in the integration, by showing the input representativeness values on the x-axis of I_P and I_M close, but not equal to 1, thus meaning that there is some loss of information in the integration. In I_C, the input representativeness values are equal to 1, since the datasets are completely represented, but the integration suffers from redundancy as shown by the lowest output representativeness value on the y-axis.

Extended evaluation. Table 5 summarizes the results of the experiments performed on all datasets in the benchmark. The second column reports the scenarios, and the other columns outline the measures obtained by considering the I_M, I_C, and I_P integrations. The bold values are the best ones, i.e. the closest to the point (1,1). We expect I_M to be the best integration in Scenario 1, I_C in Scenario 2, and I_P in Scenario 3. The measure performs correctly in almost all evaluations. Wrong best integrations in U1, U7 and U8 have all a very close distance to the best one. The mistakes are due to the sparse vocabularies and the low cardinalities in the second dataset.

Take-away: the representativeness scores offer a fine-grained explanation on why an integration strategy can be preferred to another one.

3.6 Efficiency

The time performace has been evaluated on integration processes involving datasets with increasing dimensionality (1K, 10K, 50K, 100K, 500K and 1M). These datasets have been obtained by applying sampling with replacement to the data contained in use case U10 (the largest one). The experiment was repeated 5 times and Figure 14 shows the average times. All embedding-based approaches show the same time performance, since they adopt the same algorithm for mapping the datasets into the vector space of embeddings and for computing the similarity. Moreover, they could not be applied to the largest datasets since they overcame the maximum time (48 hours) we fixed for the duration of the experiment.

Our approach shows the best performance in all configurations: it takes less than 2 minutes to compute the representativeness of the largest dataset. The vectorized implementation of the cosine similarity makes the embedding-based approaches fast, but for running on datasets larger than 100K entities it requires more memory than the one available in our system. The approach based on Jaccard's similarity has a poor performance since it cannot be vectorized for performing our computation. For this reason, the execution time grows quadratically with the size of the datasets.

Take-away: the developed approach is efficient in evaluating

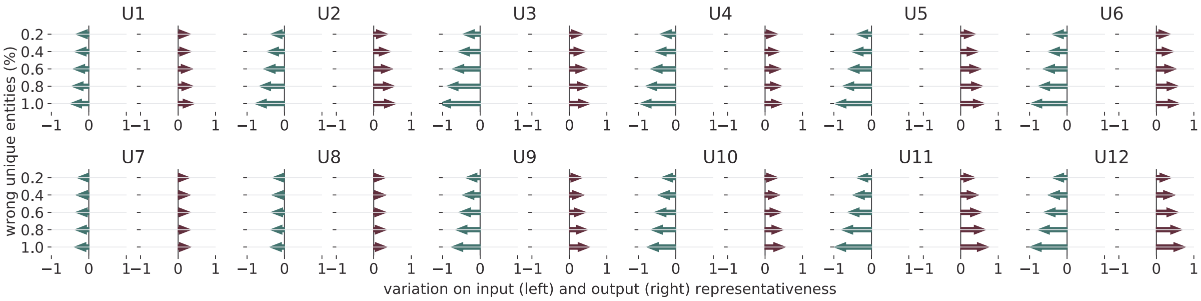


Figure 11: Representativeness variations at different unique entity error rates.

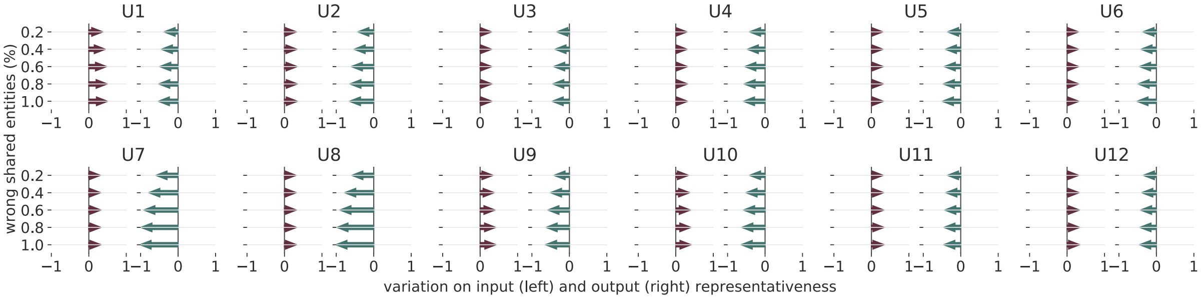
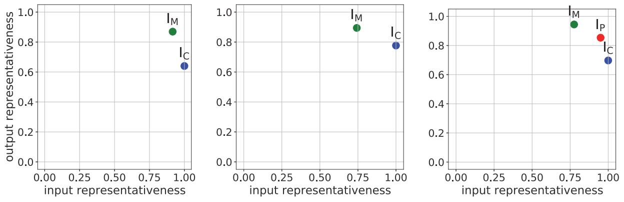


Figure 12: Representativeness variations at different sharing entity error rates.



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 13: The scenarios applied to use case U10.

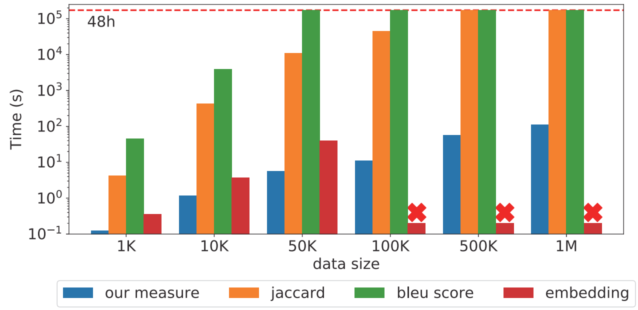


Figure 14: Computing representativeness: efficiency.

high dimensional data integration scenarios.

4. RELATED WORK

4.1 Data Integration and Entity Resolution

Data integration is one of the most challenging and long-lasting issues that the research community is confronted with for the last 30 years. The focus of the research community in the last years was mainly oriented to Entity Resolution (ER), the task concerning the development of techniques for detecting and merging entities. A number of "integration functions" to discover and match the different structures that represent the same real-world entity have been proposed [46, 64, 32, 3, 52, 49, 20, 69, 35]. Among these, rule-based and machine learning (ML) techniques are the most common ones. Regardless of the use of ML or not, ER approaches require either careful manual configuration by domain experts or a large amount of labeled data [48]. To cope with the first issue, methods have been proposed for the fine tuning of parameters such as [53], but all proposals require some human supervision. Regarding to the second problem, many semi-supervised approaches in the field of active learning [4] and crowd-sourcing [68] have been introduced. The fundamental idea behind these techniques is to limit the validation intervention required by domain experts to a minimum or to resort to crowd-workers. However, these methods suffer from a poor quality control mechanism: indeed, the former approach focuses on optimizing recall while ensuring a user-specified precision level [12, 21], while crowd-based solutions are affected by uncertain labels provided by inexperienced workers [15].

4.2 Evaluating Data Integration and Entity Resolution

The effectiveness of ER and data integration processes is typically measured against ground truths. The availability of labeled data is a problem in real scenarios, where experts have to manually assess the results obtained. This is also a problem for the evaluation of the approaches proposed by the research community since most of the techniques are evaluated against the same small number of sources

Table 5: The evaluation of the scenarios in other datasets.

Use case params	Sc.	I_M	I_C	I_P
U1 ($\lvert D1\rvert$=600, $\lvert D2\rvert$=300, $\lvert D3\rvert$=300, $\lvert D4\rvert$=600, $\lvert V1\rvert$=4776, $\lvert V2\rvert$=1431, $\lvert V3\rvert$=2258, $\lvert V4\rvert$=3092)	1	**(0.83, 0.83)**	(1.0, 0.71)	
	2	**(0.80, 0.89)**	(1.0, 0.73)	
	3	(0.7, 0.92)	(1.0, 0.73)	**(0.93, 0.77)**
U2 ($\lvert D1\rvert$=700, $\lvert D2\rvert$=350, $\lvert D3\rvert$=350, $\lvert D4\rvert$=700, $\lvert V1\rvert$=1664, $\lvert V2\rvert$=1139, $\lvert V3\rvert$=986, $\lvert V4\rvert$=1699)	1	**(0.91, 0.89)**	(1.0, 0.66)	
	2	(0.75, 0.91)	**(1.0, 0.76)**	
	3	(0.78, 0.95)	(1.0, 0.68)	**(0.92, 0.85)**
U3 ($\lvert D1\rvert$=50, $\lvert D2\rvert$=25, $\lvert D3\rvert$=25, $\lvert D4\rvert$=50, $\lvert V1\rvert$=208, $\lvert V2\rvert$=120, $\lvert V3\rvert$=136, $\lvert V4\rvert$=235)	1	**(0.9, 0.94)**	(1.0, 0.70)	
	2	(0.62, 0.97)	**(1.0, 0.89)**	
	3	(0.70, 0.97)	(1.0, 0.76)	**(0.92, 0.92)**
U4 ($\lvert D1\rvert$=100, $\lvert D2\rvert$=50, $\lvert D3\rvert$=50, $\lvert D4\rvert$=100, $\lvert V1\rvert$=375, $\lvert V2\rvert$=192, $\lvert V3\rvert$=192, $\lvert V4\rvert$=347)	1	**(0.98, 0.95)**	(1.0, 0.65)	
	2	(0.65, 0.96)	**(1.0, 0.88)**	
	3	(0.78, 0.98)	(1.0, 0.72)	**(0.98, 0.92)**
U5 ($\lvert D1\rvert$=90, $\lvert D2\rvert$=45, $\lvert D3\rvert$=45, $\lvert D4\rvert$=90, $\lvert V1\rvert$=697, $\lvert V2\rvert$=433, $\lvert V3\rvert$=462, $\lvert V4\rvert$=736)	1	**(0.92, 0.87)**	(1.0, 0.65)	
	2	(0.72, 0.93)	**(1.0, 0.79)**	
	3	(0.75, 0.94)	(1.0, 0.70)	**(0.95, 0.85)**
U6 ($\lvert D1\rvert$=90, $\lvert D2\rvert$=45, $\lvert D3\rvert$=45, $\lvert D4\rvert$=90, $\lvert V1\rvert$=503, $\lvert V2\rvert$=293, $\lvert V3\rvert$=335, $\lvert V4\rvert$=529)	1	**(0.95, 0.89)**	(1.0, 0.61)	
	2	(0.71, 0.93)	**(1.0, 0.79)**	
	3	(0.77, 0.95)	(1.0, 0.67)	**(0.98, 0.85)**
U7 ($\lvert D1\rvert$=2100, $\lvert D2\rvert$=1050, $\lvert D3\rvert$=365, $\lvert D4\rvert$=1415, $\lvert V1\rvert$=7359, $\lvert V2\rvert$=4854, $\lvert V3\rvert$=1790, $\lvert V4\rvert$=5460)	1	**(0.96, 0.87)**	(1.0, 0.59)	
	2	**(0.87, 0.79)**	(1.0, 0.7)	
	3	**(0.93, 0.91)**	(1.0, 0.61)	(0.97, 0.86)
U8 ($\lvert D1\rvert$=2100, $\lvert D2\rvert$=1050, $\lvert D3\rvert$=388, $\lvert D4\rvert$=1438, $\lvert V1\rvert$=7396, $\lvert V2\rvert$=4858, $\lvert V3\rvert$=1863, $\lvert V4\rvert$=5509)	1	**(0.98, 0.87)**	(1.0, 0.59)	
	2	**(0.87, 0.80)**	(1.0, 0.70)	
	3	**(0.93, 0.91)**	(1.0, 0.61)	(0.97, 0.86)
U9 ($\lvert D1\rvert$=2300, $\lvert D2\rvert$=1150, $\lvert D3\rvert$=1150, $\lvert D4\rvert$=2300, $\lvert V1\rvert$=5993, $\lvert V2\rvert$=4119, $\lvert V3\rvert$=3979, $\lvert V4\rvert$=6364)	1	**(0.92, 0.89)**	(1.0, 0.67)	
	2	(0.72, 0.91)	**(1.0, 0.8)**	
	3	(0.76, 0.95)	(1.0, 0.71)	**(0.93, 0.86)**
U10 ($\lvert D1\rvert$=2200, $\lvert D2\rvert$=1100, $\lvert D3\rvert$=1100, $\lvert D4\rvert$=2200, $\lvert V1\rvert$=5802, $\lvert V2\rvert$=3905, $\lvert V3\rvert$=3632, $\lvert V4\rvert$=5939)	1	**(0.92, 0.89)**	(1.0, 0.66)	
	2	(0.73, 0.90)	**(1.0, 0.79)**	
	3	(0.77, 0.95)	(1.0, 0.71)	**(0.96, 0.86)**
U11 ($\lvert D1\rvert$=700, $\lvert D2\rvert$=350, $\lvert D3\rvert$=350, $\lvert D4\rvert$=700, $\lvert V1\rvert$=2875, $\lvert V2\rvert$=2096, $\lvert V3\rvert$=1694, $\lvert V4\rvert$=3195)	1	**(0.86, 0.91)**	(1.0, 0.71)	
	2	(0.72, 0.93)	**(1.0, 0.80)**	
	3	(0.73, 0.96)	(1.0, 0.73)	**(0.88, 0.89)**
U12 ($\lvert D1\rvert$=600, $\lvert D2\rvert$=300, $\lvert D3\rvert$=300, $\lvert D4\rvert$=600, $\lvert V1\rvert$=2152, $\lvert V2\rvert$=1713, $\lvert V3\rvert$=1231, $\lvert V4\rvert$=2453)	1	**(0.88, 0.91)**	(1.0, 0.69)	
	2	(0.74, 0.96)	**(1.0, 0.79)**	
	3	(0.77, 0.95)	(1.0, 0.71)	**(0.88, 0.88)**

(typically the benchmark made available by the Magellan tool[4]) with few hundreds of labeled data. This makes possible the development and promotion of approaches overfitting on those sources (which can have features really different from the ones in sources available in real scenarios). To the best of our knowledge, only recently [41] addressed this issue, by proposing techniques for providing samples on datasets guaranteeing a fair evaluation. Similarly to other techniques [54, 28], our approach is part of this human-machine cooperation framework, but it mainly focuses on supporting analysts in the unsupervised evaluation of the integration process.

To assess the most commonly adopted evaluation framework in recent years, we survey works accepted to the main conferences and journals in the last six years. We considered scientific papers published in the following journals: IEEE Transactions on Knowledge and Data Engineering (TKDE), Journal of Data and Information Quality (JDIQ), SIGMOD Records. Furthermore, we took into consideration the subsequent international conferences: International Conference on Extending Database Technology (EDBT), International Conference on Database Theory (ICDT), IEEE International Conference on Data Engineering (ICDE), International Conference on Very Large Data Bases (VLDB), International Conference on Information and Knowledge Management-(CIKM). Among the papers selected, we consider only those containing in the title one or more of the following words: entity resolution, deduplication, matching, linkage, integration. In total, 73 papers satisfied the criteria mentioned above. We further select only papers involving some form of experimental evaluation. This further reduces the number of considered papers to 57. Table 6 reports the list of surveyed related work, divided by task addressed.

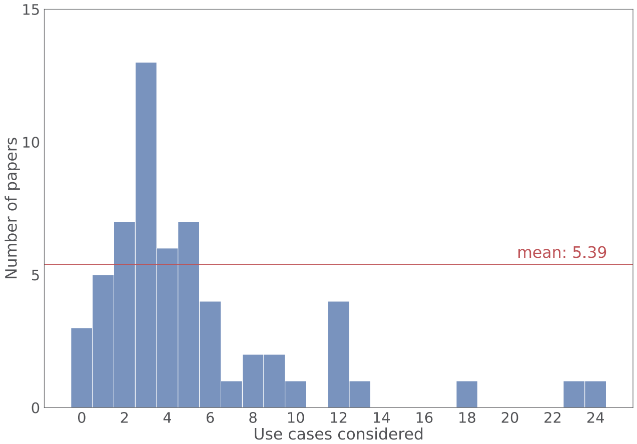


Figure 15: Number of use cases used to evaluate the proposed approaches by surveyed papers.

Figure 15 reports the distribution of the number of use cases considered by the 57 recent works in the data integration domain. The mode of considered use cases is 3, while on average 5.39 datasets are taken into account. [46, 31] present the highest number of use cases considered – respectively 23 and 24. They focused on the 24 use cases available in the Magellan repository. Among works that do not employ any publicly available dataset, they either propose a new evaluation framework [54, 32] or employ only synthetic and private datasets [27]

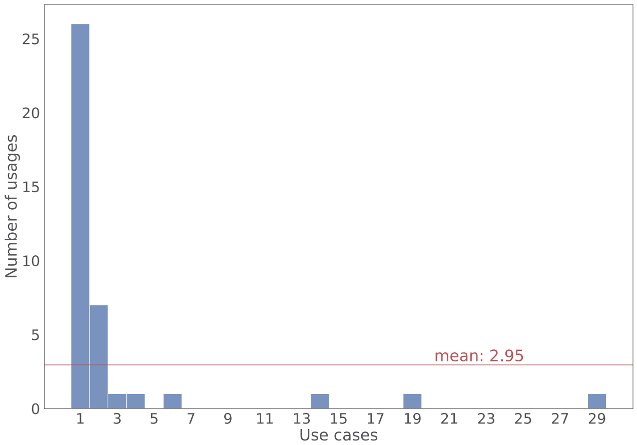


Figure 16: Number of times different use cases have been used to evaluate.

Figure 16 reports, for each use case included in the above-mentioned papers, how frequently it has been used (among the surveyed papers). In total, 26 use cases have been applied only once among the surveyed papers. This, while allowing to test each method in the most suited scenario, increases the risk of producing results that are not comparable with different works. To the other extreme, we have specific use cases, such as those derived from the Magellan

[4] `https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md`

Table 6: Surveyed related work divided by Data integration task addressed. Notice that, several works appear in more than one category.

Task	References
Schema Alignment	[10, 39, 27]
Duplicate Search Space	[30, 36, 29, 19, 24, 72, 5, 1, 70, 25, 71, 11, 13, 46, 43, 56, 62, 18]
Duplicate Recognition	[47, 30, 61, 74, 66, 26, 55, 19, 9, 8, 52, 3, 22, 12, 76, 5, 1, 70, 25, 6, 71, 67, 10, 44, 11, 13, 46, 73, 43, 41, 2, 21, 64, 18, 34, 33, 37, 58, 38, 40, 39, 63, 19, 27]
Clustering	[74, 67, 23, 2, 21, 17]
Data Fusion	[14, 60, 75, 27]
Debugging and / or complete cycle	[54, 31]

Table 7: Frequency of usage for the measures by task.

	Schema alignment	Entity resolution			Data fusion	Complete cycle
		Duplicate search space	Duplicate recognition	Clustering		
P	2	11	19	2	2	2
R	2	16	23	3	2	2
F1	3	13	36	4	1	3
AUC	—	—	1	—	—	—
MAE	—	—	1	—	1	—
Accuracy	—	—	1	—	—	—
GMD	—	—	—	1	—	—
MCC	—	—	—	—	1	—
KL divergence	—	—	—	—	1	—
Reduction Ratio	—	4	—	—	—	—
Cost	—	1	3	1	—	1
Time	1	12	15	2	2	1
Memory	—	2	—	—	—	—

repository (14 usages) and those derived from the JedAI toolkit (29 usages) that have been used in a large number of scenarios. Among the most popular use cases, the Cora Dataset [42] is one of the most popular, with 19 usages among the surveyed papers.

Concerning the measures used, the most popular evaluation approach is based on computing the F1 score. 41 papers among those surveyed adopt F1 to measure the effectiveness of the proposed approach. Similarly, 27 and 34 papers report respectively the precision and recall. Another approach often used to evaluate quantitatively the effectiveness is the *pair comparison reduction ratio*, which corresponds to the ratio between the numbers of pairs of records to be compared with and without applying a blocking strategy. The pair comparison reduction ratio is adopted in 4 of the surveyed works. A second important aspect that is measured in several works is the efficiency of the approaches. Among the considered works 23 measure the time efficiency, 4 evaluate the cost of crowd-sourcing labels to apply supervised record linkage strategies and 2 monitor the memory efficiency of the proposed strategy.

Table 7 reports measures usage by task – notice that some papers consider multiple tasks. We recognize three main categories of measures - task-agnostic effectiveness measures, task-specific effectiveness measures, and efficiency measures. In all scenarios, the most commonly used measures are those belonging to the first category, namely precision, recall and F1-score. Interestingly, depending on the task at hand, adhoc measures have been also employed. To evaluate the performance of a system on the Duplicate Search Space identification, besides task-agnostic measures, the most commonly used task-specific measure is the Reduction Ratio [26, 55, 72]. If we consider the Duplicate Recognition domain, task-specific measures used include Area Under the ROC Curve (AUC) [63], Mean Average Error (MAE) [3] and Accuracy [3]. For the Clustering task, one domain-specific measure adopted in the past is the Generalized Merged Distribution (GMD) [17]. GMD is defined as the minimal number of splits and merges required to transform the clustering into the real-world classification. Finally, when it comes to the Data Fusion tasks, considered task-specific measures include MAE [60], Matthews correlation coefficient (MCC) [14] and KL-divergence [75]. MCC is used to evaluate classification performance in a particularly unbalanced scenario [14]. Finally, KL-divergence is used to evaluate the Bayesian learning approach proposed by [75], which is evaluated by comparing the fitted probability distribution with the real one. If we consider the efficiency analyses, the most commonly evaluated aspect is the time one. This is particularly evident concerning the Duplicate Search Space identification task, with 60% of the papers that address this task analysing the temporal requirements of the proposed approach. A second aspect that is often considered is the Cost. The definition of cost slightly changes depending on the task at hand, but in general, it corresponds to the number of annotations that need to be collected to either finalize the task or construct a learning set large enough to provide sensible results.

5. CONCLUSION

We introduced the representativeness score, an unsupervised measure to evaluate the quality of an integration process by analyzing the word frequency distributions of the datasets involved. The experimental evaluation showed that the representativeness is able to provide a means for verifying and validating an integration process.

6. REFERENCES

[1] Y. Altowim and S. Mehrotra. Parallel Progressive Approach to Entity Resolution Using MapReduce. In *33rd IEEE International Conference on Data Engineering, (ICDE)*, pages 909–920, 2017.

[2] H. Altwaijry, S. Mehrotra, and D. V. Kalashnikov. QuERy: A Framework for Integrating Entity Resolution with Query Processing. *Proceedings of VLDB Endowment*, 9(3):120–131, nov 2015.

[3] A. Baraldi, F. D. Buono, M. Paganelli, and F. Guerra. Using landmarks for explaining entity matching models. In *24th International Conference on Extending Database Technology (EDBT)*, pages 451–456, 2021.

[4] K. Bellare, S. Iyengar, A. G. Parameswaran, and V. Rastogi. Active sampling for entity matching. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (KDD)*, pages 1131–1139, 2012.

[5] G. D. Bianco, R. Galante, C. A. Heuser, M. A. Gonçalves, and S. D. Canuto. A practical and effective sampling selection strategy for large scale deduplication. In *32nd IEEE International Conference on Data Engineering, (ICDE)*, pages 1518–1519, 2016.

[6] A. Bogatu, N. Paton, M. Douthwaite, S. Davie, and A. Freitas. Cost–effective Variational Active Entity Resolution. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1272–1283, 4 2021.

[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching Word Vectors with Subword Information. *ACL Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[8] U. Brunner and K. Stockinger. Entity Matching on Unstructured Data: An Active Learning Approach. In *6th Swiss Conference on Data Science, SDS*, pages 97–102, 2019.

[9] Q. Bui-Nguyen, Q. Wang, J. Shao, and D. Vatsalan. Repairing of Record Linkage: Turning Errors into Insight. In *22nd International Conference on Extending Database Technology (EDBT)*, pages 638–641, 2019.

[10] R. Cappuzzo, P. Papotti, and S. Thirumuruganathan. Creating Embeddings of Heterogeneous Relational Datasets for Data Integration Tasks. In *2020 ACM SIGMOD International Conference on Management of Data*, volume 2020, pages 1335–1349, 6 2020.

[11] C. Chai, G. Li, J. Li, D. Deng, and J. Feng. Cost-Effective Crowdsourced Entity Resolution: A Partial-Order Approach. In *2016 International Conference on Management of Data (SIGMOD)*, pages 969–984, 2016.

[12] Z. Chen, Q. Chen, F. Fan, Y. Wang, Z. Wang, Y. Nafa, Z. Li, H. Liu, and W. Pan. Enabling Quality Control for Entity Resolution: A Human and Machine Cooperation Framework. In *34th IEEE International Conference on Data Engineering, (ICDE)*, pages 1156–1167, 2018.

[13] S. Das, P. S. G. C., A. Doan, J. F. Naughton, G. Krishnan, R. Deep, E. Arcaute, V. Raghavendra, and Y. Park. Falcon: Scaling Up Hands-Off Crowdsourced Entity Matching to Build Cloud Services. In *2017 ACM International Conference on Management of Data (SIGMOD)*, pages 1431–1446, 2017.

[14] D. Deng, W. Tao, Z. Abedjan, A. K. Elmagarmid, I. F. Ilyas, G. Li, S. Madden, M. Ouzzani, M. Stonebraker, and N. Tang. Unsupervised String Transformation Learning for Entity Consolidation. In *35th IEEE International Conference on Data Engineering, (ICDE)*, pages 196–207, 2019.

[15] M. Dolatshah, M. Teoh, J. Wang, and J. Pei. Cleaning Crowdsourced Labels Using Oracles For Statistical Classification. *Proceedings of the VLDB Endowment*, 12(4):376–389, 2018.

[16] X. L. Dong and D. Srivastava. *Big Data Integration*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2015.

[17] U. Draisbach, P. Christen, and F. Naumann. Transforming Pairwise Duplicates to Entity Clusters for High-quality Duplicate Detection. *Journal of Data and Information Quality*, 12(1):1–30, 1 2020.

[18] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang. Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment*, 11(11):1454–1467, 7 2018.

[19] V. Efthymiou, G. Papadakis, K. Stefanidis, and V. Christophides. Simplifying Entity Resolution on Web Data with Schema-Agnostic, Non-Iterative Matching. In *34th IEEE International Conference on Data Engineering, (ICDE)*, pages 1296–1299, 2018.

[20] N. Fanizzi, C. d'Amato, and F. Esposito. Composite ontology matching with uncertain mappings recovery. *ACM SIGAPP Applied Computing Review*, 11(2):17–29, mar 2011.

[21] D. Firmani, B. Saha, and D. Srivastava. Online entity resolution using an Oracle. *Proceedings of the VLDB Endowment*, 9(5):384–395, 1 2016.

[22] M. Franke, Z. Sehili, F. Rohde, and E. Rahm. Evaluation of Hardening Techniques for

Privacy-Preserving Record Linkage. In *24th International Conference on Extending Database Technology (EDBT)*, pages 289–300, 2021.

[23] S. Galhotra, D. Firmani, B. Saha, and D. Srivastava. Robust Entity Resolution using Random Graphs. In *2018 International Conference on Management of Data (SIGMOD)*, pages 3–18, 2018.

[24] L. Gazzarri and M. Herschel. Boosting Blocking Performance in Entity Resolution Pipelines: Comparison Cleaning using Bloom Filters. In *23rd International Conference on Extending Database Technology (EDBT)*, pages 419–422, 2020.

[25] L. Gazzarri and M. Herschel. End-to-end Task Based Parallelization for Entity Resolution on Dynamic Data. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1248–1259, 4 2021.

[26] A. L. Gentile, P. Ristoski, S. Eckel, D. Ritze, and H. Paulheim. Entity Matching on Web Tables: a Table Embeddings approach for Blocking. In *20th International Conference on Extending Database Technology (EDBT)*, pages 510–513, 2017.

[27] B. Gu, Z. Li, X. Zhang, A. Liu, G. Liu, K. Zheng, L. Zhao, and X. Zhou. The Interaction Between Schema Matching and Record Matching in Data Integration. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):186–199, 1 2017.

[28] B. Hou, Q. Chen, Z. Chen, Y. Nafa, and Z. Li. r-HUMO: A Risk-Aware Human-Machine Cooperation Framework for Entity Resolution with Quality Guarantees. *IEEE Transactions on Knowledge and Data Engineering*, 32(2):347–359, 2020.

[29] D. Karapiperis and V. Verykios. Load-Balancing the Distance Computations in Record Linkage. *ACM SIGKDD Explorations Newsletter*, 17(1):1–7, 9 2015.

[30] A. R. Khan and H. Garcia-Molina. Attribute-based Crowd Entity Resolution. In *25th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 549–558, 2016.

[31] P. Konda, S. Das, P. S. G. C., A. Doan, A. Ardalan, J. R. Ballard, H. Li, F. Panahi, H. Zhang, J. F. Naughton, S. Prasad, G. Krishnan, R. Deep, and V. Raghavendra. Magellan: Toward Building Entity Matching Management Systems. *Proceedings of the VLDB Endowment*, 9(12):1197–1208, 2016.

[32] P. Konda, S. S. Seshadri, E. Segarra, B. Hueth, and A. Doan. Executing entity matching end to end: A case study. In *22nd International Conference on Extending Database Technology (EDBT)*, 2019.

[33] I. K. Koumarelas, T. Papenbrock, and F. Naumann. MDedup: Duplicate Detection with Matching Dependencies. *Proceedings of the VLDB Endowment*, 13(5):712–725, 2020.

[34] S. Kwashie, J. Liu, J. Li, L. Liu, M. Stumptner, and L. Yang. Certus: An Effective Entity Resolution Approach with Graph Differential Dependencies (GDDs). *Proceedings of the VLDB Endowment*, 12(6):653–666, 2019.

[35] L. Leitão and P. Calado. An automatic blocking strategy for xml duplicate detection. *ACM SIGAPP Applied Computing Review*, 13(2):42–53, jun 2013.

[36] H. Li, P. Konda, P. S. G. C., A. Doan, B. Snyder, Y. Park, G. Krishnan, R. Deep, and V. Raghavendra. MatchCatcher: A Debugger for Blocking in Entity Matching. In *21st International Conference on Extending Database Technology (EDBT)*, pages 193–204, 2018.

[37] Y. Li, J. Li, Y. Suhara, A. Doan, and W.-C. Tan. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment*, 14(1):50–60, 9 2020.

[38] Y. Li, J. Li, Y. Suhara, J. Wang, W. Hirota, and W. Tan. Deep Entity Matching: Challenges and Opportunities. *ACM Journal of Data and Information Quality*, 13(1):1:1–1:17, 2021.

[39] Y. Lin, H. Wang, J. Li, and H. Gao. Efficient Entity Resolution on Heterogeneous Records. *IEEE Transactions on Knowledge and Data Engineering*, 32(5):912–926, 2020.

[40] M. Loster, I. Koumarelas, and F. Naumann. Knowledge Transfer for Entity Resolution with Siamese Neural Networks. *ACM Journal of Data and Information Quality*, 13(1):1–25, 1 2021.

[41] N. G. Marchant and B. I. P. Rubinstein. In Search of an Entity Resolution OASIS: Optimal Asymptotic Sequential Importance Sampling. *Proceedings of the VLDB Endowment*, 10(11):1322–1333, 2017.

[42] A. McCallum. Cora Dataset, 2017.

[43] V. V. Meduri, L. Popa, P. Sen, and M. Sarwat. A Comprehensive Benchmark Framework for Active Learning Methods in Entity Matching. In *2020 International Conference on Management of Data, (SIGMOD)*, pages 1133–1147, 2020.

[44] Z. Miao, Y. Li, and X. Wang. Rotom: A Meta-Learned Data Augmentation Framework for Entity Matching, Data Cleaning, Text Classification, and Beyond. In *2021 International Conference on Management of Data (SIGMOD)*, pages 1303–1316, 2021.

[45] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ((ICLR)*, 2013.

[46] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra. Deep Learning for Entity Matching: A Design Space Exploration. In *2018 International Conference on Management of Data (SIGMOD)*, pages 19–34, 2018.

[47] H. Nie, X. Han, B. He, L. Sun, B. Chen, W. Zhang, S. Wu, and H. Kong. Deep Sequence-to-Sequence Entity Matching for Heterogeneous Entity Resolution. In *28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 629–638, 2019.

[48] S. Ortona, V. V. Meduri, and P. Papotti. Robust discovery of positive and negative rules in knowledge bases. In *34th IEEE International Conference on Data Engineering, (ICDE)*, pages 1168–1179, 2018.

[49] M. Paganelli, F. D. Buono, A. Baraldi, and F. Guerra. Analyzing how BERT performs entity matching. *Proceedings of the VLDB Endowment*, 15(8):1726–1738, 2022.

[50] M. Paganelli, F. D. Buono, F. Guerra, and N. Ferro. Unsupervised Evaluation of Data Integration Processes. In *22nd International Conference on Information Integration and Web-based Applications & Services (iiWAS)*, pages 77–81, 2020.

[51] M. Paganelli, F. D. Buono, F. Guerra, and N. Ferro. Evaluating the integration of datasets. In *SAC '22: The 37th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, April 25 - 29, 2022*, pages 347–356, 2022.

[52] M. Paganelli, F. D. Buono, M. Pevarello, F. Guerra, and M. Vincini. Automated machine learning for entity matching tasks. In *24th International Conference on Extending Database Technology (EDBT)*, pages 325–330, 2021.

[53] M. Paganelli, P. Sottovia, F. Guerra, and Y. Velegrakis. TuneR: Fine Tuning of Rule-based Entity Matchers. In *28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2945–2948, 2019.

[54] F. Panahi, W. Wu, A. Doan, and J. F. Naughton. Towards Interactive Debugging of Rule-based Entity Matching. In *20th International Conference on Extending Database Technology (EDBT)*, pages 354–365, 2017.

[55] G. Papadakis, G. Papastefanatos, T. Palpanas, and M. Koubarakis. Scaling Entity Resolution to Large, Heterogeneous Data with Enhanced Meta-blocking. In *19th International Conference on Extending Database Technology (EDBT)*, pages 221–232, 2016.

[56] G. Papadakis, J. Svirsky, A. Gal, and T. Palpanas. Comparative analysis of approximate blocking techniques for entity resolution. *Proceedings of the VLDB Endowment*, 9(9):684–695, 5 2016.

[57] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[58] R. Peeters and C. Bizer. Dual-Objective Fine-Tuning of BERT for Entity Matching. *Proceedings of the VLDB Endowment*, 14(10):1913–1921, 2021.

[59] J. Pennington, R. Socher, and C. D. Manning. Glove: Global Vectors for Word Representation. In *2014 Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, pages 1532–1543, 2014.

[60] R. Pradhan, S. Bykau, and S. Prabhakar. Staging User Feedback toward Rapid Conflict Resolution in Data Fusion. In *2017 ACM International Conference on Management of Data (SIGMOD)*, pages 603–618, 2017.

[61] K. Qian, L. Popa, and P. Sen. Active Learning for Large-Scale Entity Resolution. In *26th ACM on Conference on Information and Knowledge Management (CIKM)*, volume 2017, pages 1379–1388, 11 2017.

[62] G. Simonini, S. Bergamaschi, and H. V. Jagadish. BLAST: a Loosely Schema-aware Meta-blocking Approach for Entity Resolution. *Proceedings of the VLDB Endowment*, 9(12):1173–1184, 2016.

[63] G. Simonini, G. Papadakis, T. Palpanas, and S. Bergamaschi. Schema-Agnostic Progressive Entity Resolution. *IEEE Transactions on Knowledge and Data Engineering*, 31(6):1208–1221, 2019.

[64] R. Singh, V. V. Meduri, A. K. Elmagarmid, S. Madden, P. Papotti, J. Quiané-Ruiz, A. Solar-Lezama, and N. Tang. Synthesizing Entity Matching Rules by Examples. *Proceedings of the VLDB Endowment*, 11(2):189–202, 2017.

[65] N. A. Smith. Contextual word representations: putting words into computers. *Communications of the ACM*, 63(6):66–74, 2020.

[66] K. S. Teong, L. Soon, and T. T. Su. Schema-Agnostic Entity Matching using Pre-trained Language Models. In *29th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2241–2244, 2020.

[67] V. Verroios and H. Garcia-Molina. Entity Resolution with crowd errors. In *31st IEEE International Conference on Data Engineering, (ICDE)*, pages 219–230, 2015.

[68] N. Vesdapunt, K. Bellare, and N. N. Dalvi. Crowdsourcing algorithms for entity resolution. *Proceedings of the VLDB Endowment*, 7(12):1071–1082, 2014.

[69] A. Walker, T. Cerny, and E. Song. Open-source tools and benchmarks for code-clone detection: Past, present, and future trends. *ACM SIGAPP Applied Computing Review*, 19(4):28–39, jan 2020.

[70] H. Wang, X. Ding, J. Li, and H. Gao. Rule-based Entity Resolution on Database with hidden temporal Information. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2018.

[71] P. Wang, W. Zheng, J. Wang, and J. Pei. Automating Entity Matching Model Development. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1296–1307, 4 2021.

[72] Q. Wang, M. Cui, and H. Liang. Semantic-aware blocking for entity resolution. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 166–180, 5 2016.

[73] R. Wu, S. Chaba, S. Sawlani, X. Chu, and S. Thirumuruganathan. ZeroER: Entity Resolution using Zero Labeled Examples. In *2020 International Conference on Management of Data (SIGMOD)*, pages 1149–1164, 2020.

[74] V. Yalavarthi, X. Ke, and A. Khan. Select Your Questions Wisely. In *26th ACM International Conference on Information and Knowledge Management (CIKM)*, 11 2017.

[75] B. Zhang, S. Sanner, M. Bouadjenek, and S. Gupta. Bayesian Networks for Data Integration in the Absence of Foreign Keys. *IEEE Transactions on Knowledge and Data Engineering*, 32(4):803–808, 4 2020.

[76] D. Zhang, L. Guo, X. He, J. Shao, S. Wu, and H. T. Shen. A Graph-Theoretic Fusion Framework for Unsupervised Entity Resolution. In *34th IEEE International Conference on Data Engineering, (ICDE)*, pages 713–724, 2018.

ABOUT THE AUTHORS:

Francesco Del Buono is a PhD Candidate at the Department of Engineering “Enzo Ferrari” from the University of Modena and Reggio Emilia. His main research interests include scalable data integration, (he is working on the development of Machine Learning and Deep Learning explainable techniques to the Entity Resolution problem), and data mining and analytics on big data (he is working on techniques for anomaly and novelty detection, for supporting machine learning techniques in production).

Guglielmo Faggioli is a Research Fellow at the University of Padova. His PhD dissertation concerned the statistical modeling of systems' performance in the Information Retrieval domain and how to use such models to predict a system's performance, before its deployment. His main research interests involve information retrieval evaluation, with a focus on statistical comparison between systems, conversational search and query performance prediction.

Matteo Paganelli is a research fellow at the University of Modena and Reggio Emilia. He received the PhD degree in Computer Science from the University of Modena and Reggio Emilia in 2021, and his research area of expertise includes Big Data Management & Analytics and Big Data Integration/Entity matching.

Andrea Baraldi, PhD Candidate, Department of Engineering “Enzo Ferrari”, University of Modena and Reggio Emilia.

His main research interests include eXplainable Artificial Intelligence (XAI), Data Integration (studying explainability tools and intrinsically interpretable ML/DL models in the Entity Matching field), scalability and fairness (studying how to improve time/data efficiency of models that respect fairness constraints).

Francesco Guerra is a Full Professor at the Department of Engineering “Enzo Ferrari” from the University of Modena and Reggio Emilia, where he teaches Software Engineering and Big Data Analysis. His main research interests include scalable data management and integration, (he is working on the development of Machine Learning and Deep Learning explainable techniques to the Entity Resolution problem), semantic web (he is involved in projects for the creation and alignment of ontologies), keyword based searches on structured datasets and data mining and analytics on big data (he is working on techniques for anomaly and novelty detection, for supporting machine learning techniques in production).

Nicola Ferro is full professor in computer science at the Department of Information Engineering of the University of Padua, where he teaches information retrieval, databases, and web applications. His research interests include information retrieval, its experimental evaluation, multilingual information access and digital libraries. He is the coordinator of the CLEF evaluation initiative and the ESSIR summer school series. He has published about 400 papers on information retrieval, digital libraries, and their evaluation. More about his research at http://www.dei.unipd.it/~ferro/.