Unsupervised Evaluation of Data Integration Processes

Matteo Paganelli
Francesco Del Buono
Francesco Guerra
firstname.lastname@unimore.it
DIEF-UNIMORE

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

ABSTRACT

Evaluation of the quality of data integration processes is usually performed via manual onerous data inspections. This task is particularly heavy in real business scenarios, where the large amount of data makes checking all the tuples infeasible and the frequent updates, i.e. changes in the sources and/or new sources, impose to repeat the evaluation over and over. 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 and whether deviations are happening and a manual inspection is needed. We also conducted some preliminary experiments, using shared datasets, that show the effectiveness of the proposed measures in typical data integration scenarios.

CCS CONCEPTS

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

ACM Reference Format:
Matteo Paganelli, Francesco Del Buono, Francesco Guerra, and Nicola Ferro. 2020. Unsupervised Evaluation of Data Integration Processes. In *The 22nd International Conference on Information Integration and Web-based Applications & Services (iiWAS '20), November 30-December 2, 2020, Chiang Mai, Thailand.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3428757.3429129

1 INTRODUCTION

The massive diffusion of data analysis tools and the large training data required by machine learning applications have greatly stimulated the demand for raw data. In response, institutions and scientific communities have published a large number of open datasets, typically in the form of simple CSV-like text files, which we will refer to as *flat datasets*. This availability, however, brings a huge fragmentation of a knowledge domain, due to the many different choices operated in each dataset to represent the same domain, and it poses serious challenges when it comes to integrate many datasets into a unified knowledge representation for that domain.

iiWAS '20, November 30-December 2, 2020, Chiang Mai, Thailand

ACM ISBN 978-1-4503-8922-8/20/11…$15.00
https://doi.org/10.1145/3428757.3429129

Therefore, data integration is one of the most challenging and long-lasting issues that the research community is confronted with for the last 30 years [6]. This task is typically addressed via a *try and error* approach where a candidate integrated source is created by means of software applications and domain experts are then required to evaluate its quality and correctness. The evaluation, then, leads to modifications to the software and to the creation of a new improved release of the integrated source, that is subject to further evaluation. This process is iterated until the experts are satisfied with the result obtained.

Traditionally the creation of an integrated data source is performed by software applications implementing a pipelined architecture, which consists of three major steps: schema alignment, entity deduplication and data fusion [5]. Flat datasets typically describe just a single entity-type and, therefore, they can be integrated by using a simplified procedure where a common schema is obtained by unifying and aligning all the attributes from the different datasets and raw tuples are integrated by using Entity Resolution (ER) techniques.

ER has recently attracted marked interest by the research community and reliable approaches and tools are now available to domain experts. From the user perspective, all these approaches are black boxes providing an integration function, e.g. a set of rules or a machine learning model, "embedding" the knowledge provided by domain experts or automatically inferred by examples.

We can use measures like accuracy, or error rate, to evaluate the extent to which the integrated data source represents the original data sources. Nevertheless, the quality of the integrated source can only be assessed by an expert who has to perform the time-consuming task of manually inspecting the results of the integration process. This process becomes particularly heavy in real business scenarios, where the large amount of data makes checking all tuples infeasible. Moreover, in the case of evolving sources, where the content of the sources to integrate changes over time, and the integrated data source has to be kept aligned with them, or in case of an incremental integration process, where new sources are added over time, the evaluation has to be performed over and over.

Therefore, it is crucial to support users and domain experts with effective tools that facilitate and make the evaluation process less demanding. Our idea is to introduce an unsupervised measure of the extent to which a source "represents" the content of another source. The more a dataset can be represented by an integrated source, the less there is loss of information when the integrated source is considered in place of the original one. Ideally, a dataset should be completely represented in the integrated source. Nevertheless, loss of information is intrinsic in the process, due to the reconciliation of data inconsistencies. On the other hand, the more an integrated source is represented by an input source, the less

the integration contains duplicated data or data related to entities that are not described in the original dataset. We capture this intuition by considering the frequency distribution of the words in the sources and by quantifying how much a source is represented in another source in terms of how much their frequency distributions overlap. Experts can then use our measure to decide whether the quality of the integration is (still) good enough and no further actions are needed and/or whether deviations are happening and (an onerous) manual inspection is needed. Overall, being unsupervised, our measure does not require any additional effort by experts and can help them in reducing the amount of manual work needed.

The main contributions of the paper are: (1) the exploitation of word frequencies in an integration process as a means for measuring the quality of the integration; (2) the experimentation of the measure in typical scenarios that highlight how it can support the integration process.

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

2 RELATED WORK

Despite the number of proposed approaches, Entity Resolution (ER) is still an open challenge in the literature. A number of "integration functions" to discover and match the different structures that represent the same real-world entity has been proposed [16]. 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 [8, 10]. To cope with the first issue, methods have been proposed for the fine tuning of parameters such as [11], but all proposals require some human supervision. Regarding to the second problem, the effectiveness of ER processes is typically measured against ground truths and by using precision, recall and f-measure as metrics. The availability of labeled data is a problem in real scenarios, where experts are required to assess the results obtained. In case of large datasets, sampling techniques have to be adopted to reduce the manual evaluation. The effort required to create labeled datasets can also represent 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 (typically the benchmark made available by the Magellan tool[1]) 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 [9] addressed this issue, by proposing techniques for providing samples on datasets guaranteeing a fair evaluation. To deal with the demand of a large amount of labeled data, many semi-supervised approaches in the field of active learning and crowd-sourcing have been also introduced [2, 15]. 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 [3], while crowd-based solutions are affected by uncertain labels provided by inexperienced workers [4]. Similarly to other techniques [3, 7, 12, 17], 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.

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

3 THE APPROACH

3.1 Definitions

We consider a dataset (or source) D as a collection of entities $D = \{e_1, \ldots, e_N\}$ whose attributes are defined over a common schema $R = \{A_1, \ldots, A_M\}$; each attribute represents a specific property of an entity. The integration of datasets is performed by means of an entity resolution function, defined below.

Definition 3.1 (Entity Resolution (ER)). ER is a function that creates an integrated dataset $I = ER(\mathcal{D})$ from a collection of datasets $\mathcal{D} = \{D_1, \ldots, D_k\}$, which share a common schema R. The ER operator defines the logic for matching and merging the entities in the input dataset collection $\mathcal{D}$.

We tackle the problem of evaluating the integration process by considering how much the word frequency distribution of a dataset is "representative" of that of another dataset. Thus, the following definition introduces the concept of word frequency distribution.

Definition 3.2 (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, which we adopt in this paper, for the definition of a vocabulary of terms V for a dataset is to apply a tokenization algorithm to the concatenation of all the tuples in D. Token splitting can be considered as a solved problem [14] and a large number of techniques are available in NLP code libraries. We can now introduce the notion of "representativeness" between two datasets.

Definition 3.3 (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.

3.2 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.

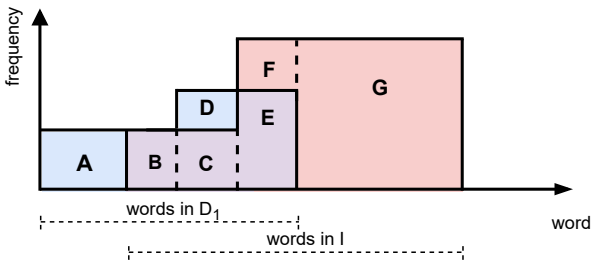


Figure 1: Example of word distributions

In the former case, 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 $r_{D\to I}$ and it is defined in equation (1).

In the latter case, we expect that an integrated dataset contains more entities than an input dataset, due to the contribution of other 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 $r_{I\to D}$ and it is defined in equation (2).

Definition 3.4. Given two datasets D and I, where I is the integration of D according to some *ER* 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 scores

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

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

We can observe as both $r_{D\to I}$ and $r_{I\to D}$ are defined over the vocabulary V_D of the dataset D and not also 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. $r_{D\to I}$, or how much it represents the integration I, i.e. $r_{I\to D}$, but without skewing the scores due to the terms of V_I which would bring in other sources than D and which may differ a lot from V_D depending on the number of integrated sources.

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 the respective distribution. Note that for sake of simplicity the heights of the frequency histograms are approximated to three possible values and the words are not marked on the x-axis. In this way, 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 represented by B, C, D, E, F are common to both the sources and: (1) the ones in B have the same frequency distribution; (2) the ones in C, D have frequency distribution C in the integration and frequency distribution $C+D$ in the input dataset; (3) the ones in E, F have frequency distribution E in the input dataset and frequency distribution $E + F$ in the integration. According to this figure, the representativeness scores are proportional to $r_{D\to I} \propto 1 - (A + \frac{D}{C+D})$, and $r_{I\to D} \propto 1 - (\frac{F}{E+F})$. In other words, the representativeness increases when the common words in D and I have the same frequency distribution.

The quality of an integration process can be evaluated by plotting the representativeness scores in a two-dimensional Cartesian plane where the x-axis reports $r_{D\to I}$ and shows the values obtained by the datasets with respect to the integration and, vice-versa, the y-axis reports $r_{I\to D}$ and shows the behavior of the integration with respect to the input sources. Values closest to the point $(1, 1)$ represent the best performance. Therefore, the more we depart from (1, 1), the more a manual inspection may be needed.

Example 2. Let us suppose to integrate the restaurant entities in datasets D_1 and D_2 in Figure 2. The vocabulary V_{D_1} is composed of 23 terms, among them, *Madison* and *WI* are the most frequent, appearing 4 times in the dataset. I_M is a possible integration, created by taking the values from D_1 in case of restaurants described in both sources; note that I_M includes, by construction, every restaurant in D_1 and D_2. As an alternative integration, consider I_C, which is obtained by simply concatenating D_1 and D_2 and thus contains duplicated restaurants.

Figure 3 shows the values of the representativeness scores obtained when datasets D_1 and D_2 in Figure 2 are integrated either in I_M or in I_C. The plot shows that I_M creates an integration that better represents the input datasets since points I_M-D_1 and I_M-D_2 in the figure are closer to 1. Considering I_C, the $r_{D\to I}$ value on the x-axis is 1 since both datasets are completely included in the integration. Nevertheless, I_C contains duplicated data, as the decreased $r_{I\to D}$ value on the y-axis shows. The reason for this behavior is the doubling of the frequencies of the duplicated words which contributes to the increase of the denominator in Equation (2).

4 SCENARIOS AND EXPERIMENTS

We assess the behavior of the representativeness scores in three typical integration scenarios. We firstly show how we create the sample datasets and we analyze the results obtained in the dataset DBLP-Google Scholar from the benchmark by the Magellan tool. Then we evaluate the other datasets published in the same benchmark by using the same approach.

Creating the datasets Firstly, the data source is deduplicated, by pairwise evaluating the entries and removing those with a Jaccard similarity greater than a specified threshold. With the remaining data, we generate three datasets, D_1, D_2, and D_3. $|D_1|$ 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. Regarding DBLP-Google Scholar, the cardinality of $|D_1|$ is 2,000 entities; the cardinality of $|D_2|$ and $|D_3|$ is 1,000 entities. The cardinality of the vocabularies associated with the datasets is $|V_1| = 5,802$, $|V_2| = 3,905$, $|V_3| = 3,632$, $|V_4| = 5,939$.

Dataset D_1	
Name	**Address**
Tutto Pasta Madison	5614 Schroeder Rd, Madison, WI
Casa Del Sol	3040 Cahill Main, Fitchburg, WI
AJ Bombers	201 W Gorham St, Madison, WI
Flaming Wok	4237 Lien Rd Ste H, Madison, WI
Dataset D_2	
Name	**Address**
Tutto Pasta Grill & Bar	5614 Schroeder Rd, Madison, WI 53711
Casa Del Sol	3040 Cahill Main, Fitchburg, WI
Acquerello	1722 Sacramento Street, San Francisco, CA
Pampas Grill	6333 W 3rd St, Los Angeles, CA
Integration I_M	
Name	**Address**
Tutto Pasta Madison	5614 Schroeder Rd, Madison, WI
Casa Del Sol	3040 Cahill Main, Fitchburg, WI
AJ Bombers	201 W Gorham St, Madison, WI
Flaming Wok	4237 Lien Rd Ste H, Madison, WI
Acquerello	1722 Sacramento Street, San Francisco, CA
Pampas Grill	6333 W 3rd St, Los Angeles, CA

Figure 2: The sources in the example

Scenario	$r_{D\to I}$	$r_{I\to D}$
$D_1 \to I_C$	1	0.729
$D_2 \to I_C$	1	0.737
$D_1 \to I_M$	1	0.964
$D_2 \to I_M$	0.891	0.932

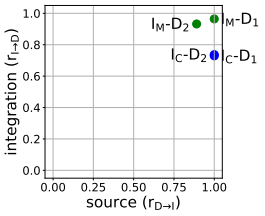
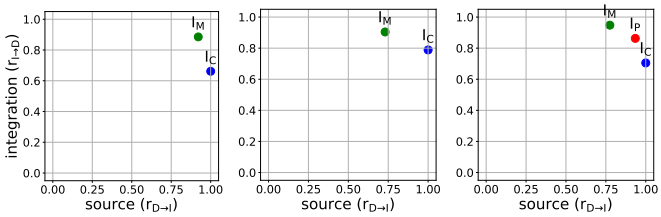


Figure 3: $r_{D\to I}$ and $r_{I\to D}$ values for the sources of Figure 2

Scenario 1: Datasets describing the same entities. We consider D_1 and D_2. Since D_1 is a superset of D_2, it can be considered as a possible integration, called $I_M = D_1$ in Figure 4a. I_C is the integration obtained by a concatenation of the tuples in D_1 and D_2. In this case 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, from a dataset perspective, the concatenation I_C is the best integration scenario, since it does not generate any loss of information. This is clear in Figure 4a, where I_C assumes the maximum value of representativeness for $r_{D\to I}$ 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 a $r_{I\to D}$ 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. Therefore, we can see how there is a general agreement between the error rate and our measures which, however, offer two advantages: (i) being unsupervised, they do not need ground-truth to be computed; (ii) they offer a more fine-grained explanation on why an integration is preferable to another one.

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 4b. 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



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

Figure 4: The scenarios used in the experiments

I_M and 0 for I_C. This is also clear by our measure, comparing the coordinates of I_C and I_M in the Figure. I_C has coordinates (1, 0.79). This means a maximum representativeness value for the sources in the integration (i.e., $r_{D\to I}$). I_M has coordinates (0.73,0.9). The $r_{D\to I}$ 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 a 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 4c 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 4c shows our measures and correctly reflects the datasets included in the integration, by showing the $r_{D\to I}$ 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 $r_{D\to I}$ values are equal to 1, since the datasets are completely represented, but the integration suffers from redundancy as shown by the lowest $r_{I\to D}$ value on the y-axis.

Extended evaluation. Table 1 summarizes the results of the experiments performed on the other datasets in the benchmark. The first column shows the names of the dataset and the cardinalities of the entities and vocabularies. 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 for each dataset in each scenario, i.e. the closest to the point (1,1). According to the previous discussion, 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 reported in the second, fourth and last dataset have all distance very close to the best one. Their mistakes are due to the sparse vocabularies (and the low cardinalities in the second dataset).

5 CONCLUSION AND FUTURE WORK

We proposed an unsupervised measure to evaluate the quality of an integration process by analyzing the word frequency distributions

Dataset	Sc.	I_M	I_C	I_P
Structured Walmart-Amazon	1	**(0.88, 0.91)**	(1.0, 0.69)	
(|D1|=600, |D2|=300, |D3|=300, |D4|=600,	2	(0.74, 0.96)	**(1.0, 0.79)**	
|V1|=2152, |V2|=1713, |V3|=1231, |V4|=2453)	3	(0.77, 0.95)	(1.0, 0.71)	**(0.88, 0.88)**
Structured DBLP-ACM	1	**(0.98, 0.87)**	(1.0, 0.59)	
(|D1|=2100, |D2|=1050, |D3|=388, |D4|=1438,	2	**(0.87, 0.80)**	(1.0, 0.70)	
|V1|=7396, |V2|=4858, |V3|=1863, |V4|=5509)	3	**(0.93, 0.91)**	(1.0, 0.61)	(0.97, 0.86)
Structured Beer	1	**(0.9, 0.94)**	(1.0, 0.70)	
(|D1|=50, |D2|=25, |D3|=25, |D4|=50,	2	(0.62, 0.97)	**(1.0, 0.89)**	
|V1|=208, |V2|=120, |V3|=136, |V4|=235)	3	(0.70, 0.97)	(1.0, 0.76)	**(0.92, 0.92)**
Textual Abt-Buy	1	**(0.83, 0.83)**	(1.0, 0.71)	
(|D1|=600, |D2|=300, |D3|=300, |D4|=600,	2	**(0.80, 0.89)**	(1.0, 0.73)	
|V1|=4776, |V2|=1431, |V3|=2258, |V4|=3092)	3	(0.7, 0.92)	(1.0, 0.73)	**(0.93, 0.77)**
Structured Amazon-Google	1	**(0.91, 0.89)**	(1.0, 0.66)	
(|D1|=700, |D2|=350, |D3|=350, |D4|=700,	2	(0.75, 0.91)	**(1.0, 0.76)**	
|V1|=1664, |V2|=1139, |V3|=986, |V4|=1699)	3	(0.78, 0.95)	(1.0, 0.68)	**(0.92, 0.85)**
Dirty DBLP-GoogleScholar	1	**(0.92, 0.89)**	(1.0, 0.67)	
(|D1|=2300, |D2|=1150, |D3|=1150, |D4|=2300,	2	(0.72, 0.91)	**(1.0, 0.8)**	
|V1|=5993, |V2|=4119, |V3|=3979, |V4|=6364)	3	(0.76, 0.95)	(1.0, 0.71)	**(0.93, 0.86)**
Dirty Walmart-Amazon	1	**(0.86, 0.91)**	(1.0, 0.71)	
(|D1|=700, |D2|=350, |D3|=350, |D4|=700,	2	(0.72, 0.93)	**(1.0, 0.80)**	
|V1|=2875, |V2|=2096, |V3|=1694, |V4|=3195)	3	(0.73, 0.96)	(1.0, 0.73)	**(0.88, 0.89)**
Structured iTunes-Amazon	1	**(0.95, 0.89)**	(1.0, 0.61)	
(|D1|=90, |D2|=45, |D3|=45, |D4|=90,	2	(0.71, 0.93)	**(1.0, 0.79)**	
|V1|=503, |V2|=293, |V3|=335, |V4|=529)	3	(0.77, 0.95)	(1.0, 0.67)	**(0.98, 0.85)**
Dirty iTunes-Amazon	1	**(0.92, 0.87)**	(1.0, 0.65)	
(|D1|=90, |D2|=45, |D3|=45, |D4|=90,	2	(0.72, 0.93)	**(1.0, 0.79)**	
|V1|=697, |V2|=433, |V3|=462, |V4|=736)	3	(0.75, 0.94)	(1.0, 0.70)	**(0.95, 0.85)**
Structured Fodors-Zagats	1	**(0.98, 0.95)**	(1.0, 0.65)	
(|D1|=100, |D2|=50, |D3|=50, |D4|=100,	2	(0.65, 0.96)	**(1.0, 0.88)**	
|V1|=375, |V2|=192, |V3|=192, |V4|=347)	3	(0.78, 0.98)	(1.0, 0.72)	**(0.98, 0.92)**
Dirty DBLP-ACM	1	**(0.96, 0.87)**	(1.0, 0.59)	
(|D1|=2100, |D2|=1050, |D3|=365, |D4|=1415,	2	**(0.87, 0.79)**	(1.0, 0.7)	
|V1|=7359, |V2|=4854, |V3|=1790, |V4|=5460)	3	**(0.93, 0.91)**	(1.0, 0.61)	(0.97, 0.86)

Table 1: The evaluation of the scenarios in other datasets

of the datasets involved. The preliminary experiments confirm that the measure behaves as expected in three typical scenarios. However, some improvements can (1) reduce the mismatches due to the use of a measure computed over the entire dataset to evaluate a "local" behavior (i.e., the correspondence of tuples); (2) make the approach usable with numeric datasets. To deal with the first issue, our idea is to evaluate a "local" model which compares word frequencies of the tuples in the input datasets with the ones in the integrated source. To be able to evaluate numeric datasets, our idea is to exploit functional dependencies (FD), i.e., when the values in one set of columns functionally determine the value of another column [1]. Tools like Metanome [13] can easily retrieve the FDs existing in a dataset. Our model based on word frequencies can be easily extended to a model based on FD frequencies.

REFERENCES

[1] Ziawasch Abedjan, Lukasz Golab, Felix Naumann, and Thorsten Papenbrock. 2018. *Data Profiling.* Morgan & Claypool Publishers.

[2] Arvind Arasu, Michaela Götz, and Raghav Kaushik. 2010. On active learning of record matching packages. In *SIGMOD*, Ahmed K. Elmagarmid and Divyakant Agrawal (Eds.). ACM, 783–794.

[3] Zhaoqiang Chen, Qun Chen, Fengfeng Fan, Yanyan Wang, Zhuo Wang, Youcef Nafa, Zhanhuai Li, Hailong Liu, and Wei Pan. 2018. Enabling Quality Control for Entity Resolution: A Human and Machine Cooperation Framework. In *ICDE*. 1156–1167.

[4] Mohamad Dolatshah, Mathew Teoh, Jiannan Wang, and Jian Pei. 2018. Cleaning Crowdsourced Labels Using Oracles For Statistical Classification. *Proc. VLDB Endow.* 12, 4 (2018), 376–389.

[5] Xin Luna Dong and Divesh Srivastava. 2015. *Big Data Integration.* Morgan & Claypool Publishers. https://doi.org/10.2200/S00578ED1V01Y201404DTM040

[6] Behzad Golshan, Alon Y. Halevy, George A. Mihaila, and Wang-Chiew Tan. 2017. Data Integration: After the Teenage Years. In *SIGMOD*.

[7] Francesco Guerra, Paolo Sottovia, Matteo Paganelli, and Maurizio Vincini. 2019. Big Data Integration of Heterogeneous Data Sources: The Re-Search Alps Case Study. In *BigData Congress 2019, Milan, Italy.*

[8] Lingli Li, Jianzhong Li, and Hong Gao. 2015. Rule-Based Method for Entity Resolution. *IEEE Trans. Knowl. Data Eng.* (2015).

[9] Neil G. Marchant and Benjamin I. P. Rubinstein. 2017. In Search of an Entity Resolution OASIS: Optimal Asymptotic Sequential Importance Sampling. *Proc. VLDB Endow.* 10, 11 (2017), 1322–1333. https://doi.org/10.14778/3137628.3137642

[10] Stefano Ortona, Venkata Vamsikrishna Meduri, and Paolo Papotti. 2018. Robust Discovery of Positive and Negative Rules in Knowledge Bases. In *ICDE*.

[11] Matteo Paganelli, Paolo Sottovia, Francesco Guerra, and Yannis Velegrakis. 2019. TuneR: Fine Tuning of Rule-based Entity Matchers. In *CIKM*.

[12] Fatemah Panahi, Wentao Wu, AnHai Doan, and Jeffrey F. Naughton. 2017. Towards Interactive Debugging of Rule-based Entity Matching. In *EDBT*.

[13] Thorsten Papenbrock, Tanja Bergmann, Moritz Finke, Jakob Zwiener, and Felix Naumann. 2015. Data Profiling with Metanome. *PVLDB* (2015).

[14] Noah A. Smith. 2020. Contextual Word Representations: Putting Words into Computers. *Commun. ACM* 63, 6 (May 2020), 66–74.

[15] Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. 2012. CrowdER: Crowdsourcing Entity Resolution. *Proc. VLDB Endow.* 5, 11 (2012), 1483–1494.

[16] Jiannan Wang, Guoliang Li, Jeffrey Xu Yu, and Jianhua Feng. 2011. Entity Matching: How Similar Is Similar. *PVLDB* (2011).

[17] Steven Euijong Whang and Hector Garcia-Molina. 2014. Incremental entity resolution on rules and data. *VLDB J.* (2014).