Towards Information Profiling: Data Lake Content Metadata Management

Ayman Alserafi, Alberto Abelló, Oscar Romero
Universitat Politècnica de Catalunya (UPC),
Barcelona, Catalunya, Spain
{alserafi, aabello, oromero}@essi.upc.edu

Toon Calders
Université Libre de Bruxelles (ULB), Brussels, Belgium
& Universiteit Antwerpen (UAntwerp), Antwerp, Belgium
{toon.calders} @ulb.ac.be & @uantwerp.be

***Abstract*—There is currently a burst of Big Data (BD) processed and stored in huge raw data repositories, commonly called Data Lakes (DL). These BD require new techniques of data integration and schema alignment in order to make the data usable by its consumers and to discover the relationships linking their content. This can be provided by metadata services which discover and describe their content. However, there is currently a lack of a systematic approach for such kind of metadata discovery and management. Thus, we propose a framework for the profiling of informational content stored in the DL, which we call information profiling. The profiles are stored as metadata to support data analysis. We formally define a metadata management process which identifies the key activities required to effectively handle this. We demonstrate the alternative techniques and performance of our process using a prototype implementation handling a real-life case-study from the OpenML DL, which showcases the value and feasibility of our approach.**

I. Introduction

There is currently a huge growth in the amount, variety, and velocity of data ingested in analytical data repositories. Such data are commonly called Big Data (BD). Data repositories storing such BD in their original raw-format are commonly called Data Lakes (DL) [1]. DL are characterised by having a large amount of data covering different subjects, which need to be analysed by non-experts in IT commonly called data enthusiasts [2]. To support the data enthusiast in analysing the data in the DL, there must be a data governance process which describes the content using metadata. Such process should describe the informational content of the data ingested using the least intrusive techniques. The metadata can then be exploited by the data enthusiast to discover relationships between datasets, duplicated data, and outliers which have no other datasets related to them.

In this paper, we investigate the appropriate process and techniques required to manage the metadata about the informational content of the DL. We specifically focus on addressing the challenges of variety and variability of BD ingested in the DL. The metadata discovered supports data consumers in finding the required data in the large amounts of information stored inside the DL for analytical purposes [3]. Currently, information discovery to identify, locate, integrate and reengineer data consumes 70% of time spent in data analytics project [1], which clearly needs to be decreased. To handle this challenge, this paper proposes (i) a systematic process for the schema annotation of data ingested in the DL and (ii) the systematic extraction, management and exploitation of metadata about the datasets' content and their relationship by means of existing schema matching and ontology alignment techniques [4], [5], [6].

The proposed process allows for the automation of data governance tasks for the DL. To our knowledge, the proposed framework is the first holistic approach which integrates automated techniques for supporting analytical discovery of cross-DL content relationships, which we call *information profiles* as explained below. This should cater for the current shortcoming of a formalized metadata management process to prevent the DL from becoming a *data swamp*; that is a DL that is not well governed and can not maintain appropriate data quality. Data swamps store data without metadata describing them, decreasing their utility [4].

Information Profiling. Traditional schema extraction and data profiling involves analysing raw data for detecting structural patterns and statistical distributions [7]. There is currently a need for higher-level profiling which involves analysing information about the approximate schema & instances relationships between different datasets instead of just single datasets [8], which we specifically define as *Information Profiling*. This involves the analysis of metadata and schema [8], [9] extracted from the raw data using ontology alignment techniques [5], [6]. Such techniques exploit 1. schema metadata and 2. data profile metadata to match different attributes from different datasets, generating the information profile. A schema profile describes the schema of datasets, e.g. how many attributes, their data types, and the names of the attributes [10]. The data profiles considered describe the values of the dataset, i.e. the single-attribute statistics of values [7]. Information profiles, the 3^{rd} type of content metadata, exploits the patterns from data profiles and data schemas [3]. For example, annotating attributes which can be linked based on approximate similarity of data distributions and data types.

Content Metadata. Content metadata is the representation of all types of profiles in the DL. Of our interest is augmenting metadata describing the informational content of datasets as first-class citizens, in order to support exploratory navigation of the DL. This involves representing the schema and profiles of data ingested in semantic-enabled standards like RDF [1],

[1]https://www.w3.org/RDF/

which is a recommendation of the W3C for representing metadata. It is important to have metadata in semantically-enabled formats, because it supports information profiling using schema matching and ontology alignment techniques like [5], [6].

Contributions. The main contribution here is an end-to-end content metadata management process which provides *a systematic approach for data governance*. We identify the key tasks and activities for content metadata management in the DL for alignment purposes [11]. We focus on detecting three types of relationships: duplicate datasets, related datasets (i.e. "joinable" data attributes between datasets), and outlier datasets. This includes (i) identification of what content metadata must be collected for detecting relationships between datasets. In addition, (ii) identification of methods to collect such metadata to annotate the datasets. Finally, (iii) we prove the feasibility of our approach using a prototype applied to a real-life case-study. With the challenge of new formats of raw data flowing inside the DL and the high variability of such data, the answer to these challenges is non-trivial. Difficulties here include effective techniques for sampling the data to improve efficiency, applying the right matching techniques and efficiently using them for convergence. We propose a framework catering for those challenges which considers the schema, data, and information profile metadata managed.

For the remainder of the paper: we review related work in Section II; we demonstrate our approach using a motivational case-study in Section III; we propose a framework & process for managing such metadata in Section IV; we showcase a prototype implementing our approach in Section V; we follow with results from experimenting with the prototype on the DL from the motivational example in Section VI; and we conclude the paper with a discussion of the metadata management approach and recommendations for future research in Sections VII and VIII.

II. Related Work

There is currently missing a holistic approach of informational content metadata management to support the data enthusiast [1], [2]. The DL also needs to have accompanying metadata to prevent it from becoming a data swamp [4]. Currently, data profiling and annotation is of great importance for research in DL architectures and is currently a hot topic for research [3], [12], [13]. Some techniques and approaches were previously investigated, but are mainly focused on relational content metadata [7], [10], free-text metadata [13], or data provenance metadata [1], [14]. Most of the current research efforts are suggesting the need for a governed metadata management process for integrating different varieties of BD [8], [13], [15]. This is currently handled by manual inspection of the data in the DL which consumes a lot of time and results in a big analytical latency [15]. Our proposed framework handles this metadata using automatic techniques.

Many research efforts are targeting *extraction of schema and content metadata*. Those provide an overview of techniques, algorithms and approaches to extract schemas, matching schemas, and finding patterns in the data content of data files [13], [16]. There is also research to detect **cross-data relationships** which aim at detecting similar data files with similar informational concepts [15], [17].

TABLE I
Description of OpenML datasets

Domain	Datasets IDs	Datasets
Vehicles	21,455,967,1092	car,cars,cars,Crash
Business	223,549,841	Stock,strikes,stock
Sports	214	baskball
Health	13,15,37	breast-cancer,breast-w,diabetes
Others	48,50,61,969	tae,tic-tac-toe,Iris,Iris

Ontology alignment and schema matching techniques which are based on finding similarity between data schemas and instances of data can also be utilised to integrate datasets [5]. This can be achieved by extracting the schema and ontology from the data and then applying the matching techniques [16].

The current shortcoming of research about managing metadata in the DL is that the available techniques are not formally defined as a systematic process for data governance, it is still applicable only to relational data warehouses, and does not handle the automatic annotation of informational content of datasets in the DL. We cover this gap by proposing an automatic content metadata management process. The ontology alignment techniques were also classically applied to discover similarity between two large ontologies [6], but have not been sufficiently applied before to duplicate detection, outlier detection and cross-datasets relationships extraction on multiple discrete datasets.

III. Motivational Case-Study

In order to demonstrate the feasibility and value of our systematic approach for content metadata discovery, we implement a prototype called *Content Metadata for Data Lakes (CM4DL)*. This prototype is tested with a real-life example of a DL called OpenML[2]. OpenML is a web-based data repository which allows data scientists to contribute different datasets which can be used in data mining experiments [18]. The OpenML platform supports loading different types of data which are stored in the WEKA[3] format (i.e. ARFF). OpenML stores datasets which represent diverse data domains, and can be considered a DL because it involves raw data loaded without a specific integration schema and which represent diverse subject-areas intended for analytics. A subset of this DL involving 15 datasets categorized into 5 subject-areas were used in our experiments and can be seen in Table I (it uses the OpenML dataset-ID, which can be used for retrieving the data using the OpenML API[4]. The dataset names from OpenML are given in the last column).

OpenML provides pre-computed data profiles for each dataset as JSON files (retrievable by the API too), which we have parsed in our prototype and used to compare the datasets with each other. This includes the statistical distribution of

[2]http://www.openml.org/
[3]http://www.cs.waikato.ac.nz/ml/weka/
[4]http://www.openml.org/guide

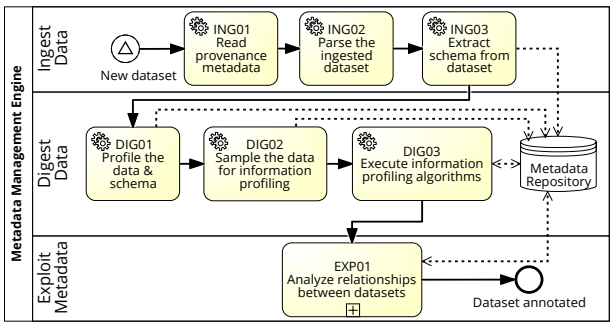

Fig. 1. The Metadata Management BPMN Process Model

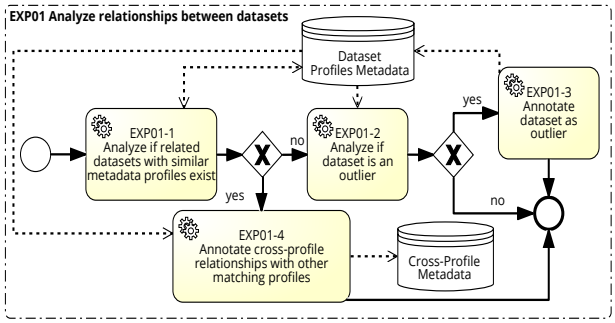

Fig. 2. The EXP01 Metadata Exploitation Sub-Process Model

numerical attributes and the value frequency distribution of nominal attributes [18]. The datasets will be used in our experiments as input datasets. They will be automatically annotated to describe their content (attributes and instances). Each dataset consists of a number of attributes for each instance. Each instance of a dataset shares the same attributes.

$$x = [d(d-1)/2] * m^2 \quad (1)$$

The number of attributes is 10 per dataset on average. In order to compare all attributes together from those datasets, there needs to be about 10500 comparisons according to Equation 1. This approximates the number of comparisons x in terms of d number of datasets and m number of attributes, not comparing a dataset to itself or to other datasets twice. This is very difficult for a human to achieve and will require a huge effort (as will be described in the experiments in Section VI). Therefore, it is important to have an automatic process which is capable of efficiently executing those comparisons and capturing the important informational relationships between the datasets. Challenges which arise include efficiently handling the large varieties of datasets in OpenML. The automated process handling this is described in the next Section.

IV. A Framework for Content Metadata Management

In this section, we propose a framework for the automatic management of metadata about the DL. The goal is a cross-datasets relationships aware DL which can be navigated easily. This framework integrates the different schema matching and ontology alignment techniques for the purpose of information profiling. Metadata annotation can be efficient and does not heavily affect processing times of datasets in the DL as shown in related experiments like [13], [14] and in our experiments in Section VI.

The framework involves 3 main phases. The first phase is **data ingestion** which includes discovering the new data by its provenance metadata, parsing the data, extracting the schema of the data similar to [16], and storing the data in the DL with its annotated *schema metadata*. The second phase is **data digestion** which means analysing the data flowing to the DL to discover informational concepts and data features. This phase includes data profiling, schema profiling and ontology alignment to extract information profiles (i.e. extraction of all *content metadata* artefacts). The datasets are annotated with their profiles in the *metadata repository*. The third phase includes **metadata exploitation** like discovering relationships between datasets. This involves information profiling which exploits the *content metadata* from the data digestion process to detect and annotate the relationships of a dataset with other related datasets which can be analysed together [3]. Those are called *cross-dataset relationship metadata*.

The framework is implemented by a structured metadata management process, which can be seen in Figure 1. This facilitates systematically collecting and maintaining the metadata throughout the lifetime of the DL. To define the activities for this framework, we present a BPMN process model. Each activity in the BPMN model is described below by the technique along with its computational complexity, and description of what is achieved.

Start & data ingestion. The dataset annotation process starts when a signal arrives to the metadata engine, indicating that a new dataset is uploaded to the DL. In ING01, the dataset is located using its provenance metadata in $O(1)$ time. Then it is parsed in ING02 to verify its structural correctness in $O(n)$ time, where n is number of instances. The dataset is then analysed in activity ING03 to extract and annotate the schema semantics in $O(m)$ time, where m is the number of attributes. This is done using RDF ontology extraction techniques like in [16]. The generated metadata is stored in a semantic-aware metadata repository (i.e. RDF Triplestore[5]).

Data digestion. The dataset is then digested to extract the content metadata. This starts by DIG01 which creates the data profile and schema profile using simple statistical techniques and profiling algorithms similar to [7]. This is done in $O(n)$ time. The following activity DIG02 samples the data instances to improve the efficiency of the information profiling algorithms in the next activity, which is completed in $O(1)$ time. In DIG03, the dataset and its profiles are compared to other datasets and their profiles using ontology alignment techniques, which requires $O(m^2)$ in worst case scenario [11]. We propose an algorithm to reduce this complexity in Section V. There should be certain cut-off thresholds of schema similarity (like [13], [16], [17]) and data profile similarity [12] to indicate whether to align two datasets together, in order to decrease the number of comparisons made in this activity. Ontology alignment is used to extract metadata about the relationships with other datasets. The existing alignment techniques we utilize first hash and index the values from the

[5]https://www.w3.org/wiki/LargeTripleStores

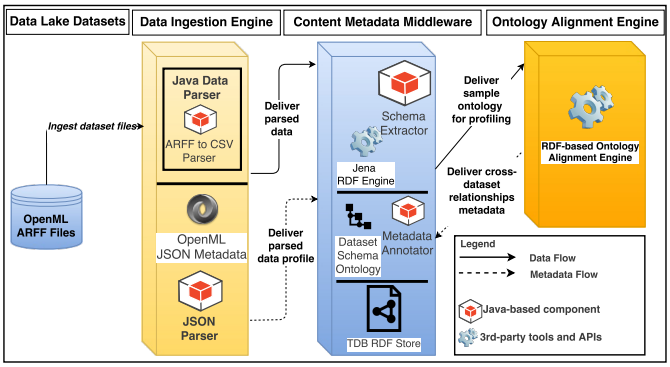

Fig. 3. CM4DL System Architecture

data instances like [19], then use an alignment algorithm like [6] to match the attributes from the datasets. The dataset is analysed to extract its information profile and then goes to the exploitation phase of the framework.

Metadata exploitation. This starts in the EXP01 subprocess which detects relationships with other datasets using the content metadata stored in the *Metadata Repository*. This can be seen in Figure 2. This includes EXP01-1 which checks if similar attributes in other datasets exist by comparing the similarity stored between datasets' attributes against a specific threshold. If the similarity of attributes exceeds the threshold then related datasets exists, and the flow follows with EXP01-4 which annotates the cross-profile relationships and stores this as the *Cross-Profile Metadata*. We also discover duplicate datasets in EXP01-4. Those are datasets with the same profiles including the same schema structure, with the same number of attributes, and similar data profile (i.e. overlapping value frequency distributions). Otherwise, if there is no related datasets detected in EXP01-1, the dataset is checked in EXP01-2 to see whether it is an outlier [10]. The dataset is an outlier if it has no matching attributes with other datasets found in the metadata repository, and is annotated as an outlier in EXP01-3.

The remainder of this paper discusses an instantiation of this process and experimental results from its implementation.

V. The CM4DL Prototype

In order to instantiate the BPMN model in Figures 1 & 2 and to prove its feasibility, we implement a prototype called *Content Metadata For Data Lakes (CM4DL in short)*. The prototype consists of multiple components and the system architecture can be seen in Figure 3. The prototype is based on a Java implementation. Tools and APIs which are developed by 3rd-parties, but which are utilised are shown using a different symbol as seen in the legend.

A. Prototype architecture

The prototype consists of three main layers, in addition to the DL dataset files. The DL files containing the datasets are first read along with their accompanying JSON metadata from OpenML (using the OpenML Java-based API library). This retrieves the ARFF datasets and JSON metadata objects from the OpenML library and stores it on the local server machine. In the data ingestion engine layer, a Java data parser component is utilised. The data parser reads the ARFF files using the WEKA Java API and converts the datasets to CSV files. This conversion is based on mapping the ARFF attributes to CSV columns. The parser utilises the JSON metadata files provided by the OpenML API which describe each attribute and its data-type. This provides us with pre-computed data-profiles describing the dataset and each attribute in the dataset. For numeric attributes, the metadata includes min, max, mean, and standard deviation. For nominal and string attributes, it provides the full frequency distribution of values.

The next layer is the main component for content metadata management, which is called Content Metadata Middleware. It is responsible for first converting the datasets from OpenML to RDF schemas. This is done using the schema extractor which loads the CSV files into a Jena TDB RDF triplestore. Those files are sampled for a specific number of instances, and are then parsed using the Jena RDF library for Java[6]. The output leads to a mapping of each ingested dataset with an RDF N-triple ontology representing the schema and its sampled instances. Each dataset is represented as an RDF *class* and each attribute as an RDF *property*. Finally, a metadata annotator uses the generated ontology mappings to discover relationships between datasets consisting of similar attributes. This matching task is done using the ontology alignment engine in the last layer which detects schema- and instance-based relationships between datasets, and returns them to the metadata annotator to be stored in the *Metadata Repository*.

Each component in the system architecture of the prototype implements and automates activities from the BPMN process in Figure. 1. The Java Data Parser handles the activities ING01 and ING02. The schema extractor in the middleware layer handles activity ING03. The data and schema profile is ingested in the OpenML JSON metadata and JSON parser which provide the profile metadata for the middleware. The metadata annotator in the middleware is able to exploit this profile metadata and the schema metadata to detect duplicates using profile querying and ontology alignment respectively. This handles activity DIG03. To detect relationships between datasets, EXP01 is implemented in the ontology alignment layer to detect related datasets and their related attributes by analysing the information profiles extracted in DIG03.

B. Ontology alignment component

In the CM4DL prototype, we utilize the existing ontology alignment engines to facilitate our approach. The field of ontology alignment is very developed and to understand the basic techniques of such tools you can refer to the following references: [5], [6], [19]. In order to select an appropriate tool for our task, we evaluated the research literature for a tool which supports the following:

- **Schema- and instance- based ontology alignment:** the tool needs to analyse both the schema (attribute types and dependencies) and the instances (values for the attribute) to check for similarity. [5] compares such techniques.

[6]https://jena.apache.org/

TABLE II
EXAMPLE CROSS-DATASET RELATIONSHIPS

No.	Dataset 1	Dataset 2	Attribute 1	Attribute 2	Relationship
1	37 (diabetes)	214 (basktball)	age	age	related
2	455 (cars)	549 (strikes)	model.year	year	related
3	455 (cars)	967 (cars)	all	all	duplicate
4	455 (cars)	1092 (Crash)	name	model	related
5	455 (cars)	1092 (Crash)	weight	Wt	related
7	50 (tic-tac-toe)	N/A	all	N/A	outlier

- **Indexing and hashing techniques (like the MinHash algorithm [19]):** this is essential to speed-up the comparison of the datasets and to make this more efficient.
- **Different techniques of instance-based similarity:** the tool should implement different techniques for comparing values of instances like different string-comparison techniques (e.g., normalised identities [6], shingling-MinHash distances [19], etc.). The different similarity comparison techniques can yield different effectiveness with different types of data. Therefore, it is important to study different comparison techniques for effectiveness and efficiency in our task.
- **Open-source Java API:** the tool must expose an open-source API which is integrable within our developed prototype.

From the short-listed tools, according to the above criteria, we identified COMA++ [5] and PARIS probabilistic ontology alignment [6] as possible candidates. We selected PARIS because of its simplicity in integration with a Java-based API and being cited for having high effectiveness with large-scale ontology alignment when compared against other tools and benchmarks (see [11]). PARIS aligns ontologies [6] by finding RDF *subclasses* which in our case indicates the similarity of datasets, and RDF *subproperties* indicating similarity of attributes in the datasets. Similarity is given as a percentage; higher values means more similar.

The ontology alignment tool is capable of reading two ontologies and detecting the degree of similarity between the two ontologies based on the schema and instances in the ontology [6]. The ontology must be defined in N-Triples[7] RDF representation. The metadata annotator component can send any two datasets in N-Triples format and then the tool will return the similarity of classes (i.e. datasets) from both ontologies (coefficient between 0 and 1) along with similarity between both datasets attributes (modelled as RDF properties). The similarity is based on comparing instances (modelled as RDF concepts) using string matching techniques. In PARIS [6], there are two techniques provided, which we have used in our prototype: the identity-based exact match [6] and the shingling-based MinHash approximate matching [19]. For the identity-based approach the attribute values are normalized by removing punctuation marks and converting the characters to lower-case. The normalized text are then compared for exact matches. This works best for numeric attributes with exact values. The shingling-based approach compares n-grams of text (i.e. specific number of character sequences) and is better suited for approximate matching of strings.

Relationships detected in this layer include examples like those in Table II which are based on the OpenML datasets used in the experiments. The table compares attributes from two datasets by showing their relationship. Each dataset is described by its OpenML ID and dataset name. The attribute name from each dataset is then given. Finally a relationship is listed as either: related, duplicate, or outlier. The relationship *related* is used to identify similar attributes which can be used to "link" the datasets together. The relationships are identified by analysing the similarity of the actual value distribution of the attributes as exhibited by the instances of data in the dataset [6]. Related attributes should have an overlapping distribution of values which can be used to link the attributes together. The ontology alignment algorithms should be capable to detect that attributes like those in relationships no. 2, 4, and 5 are related. Although the attributes have different names in the schema, their values are overlapping and hold similar character- or numeric- values. Therefore, it is important to use ontology alignment which is instance-based to detect such relationships.

In addition, when all attributes are related with attributes of another dataset we call this relationship a *duplicate* relation. This means the datasets contain similar informational content in all their attributes. This can be seen for example in Table II row 3. Detecting duplicates can help in data cleansing and de-duplication by eliminating or merging them to maintain high data quality in the DL with less redundancy. It is based on taking a cut-off threshold of similarity generated by the ontology alignment tool to indicate if datasets are duplicates (e.g. taking 0.8 for similarity of all attributes). Finally, an *outlier* is a dataset which has no related attributes in any of the other datasets in the data lake. For outliers, all attributes of a dataset have no matching attributes in any other dataset.

C. Dataset comparison algorithm

In order to match the datasets we use Algorithm 1. It automates the information profiling activity DIG03 in Figure 1, however, note that the BPMN describes the handling of each separate dataset while the algorithm describes the overall *collective* handling of datasets. The matching algorithm is based on the ontology alignment similarity measure [6] and the average data and schema profile similarity. The profile similarity is calculated as the average of the difference between the normalized profile features from each dataset. The list of profile features used includes: the number of attributes in the dataset, number and percentage of numerical/binary/symbolic attributes, the number of classes for the target variable, the size and percentage of the majority and minority classes of the target variable, the number of instances in the dataset, percentage of instances with missing values, and the dimensionality measure. This subset of features were selected because they are the most frequently occurring in the OpenML JSON metadata.

The algorithm consists of input DL N-Triple files (*DL-NTriples*), the JSON metadata features (*ProfileMetadata*), and the thresholds for matching datasets on the basis of profile metadata (*ProfileThreshold*), or thresholds for matching attributes in the ontology alignment tool as related (*Relation-Threshold*) or duplicates (*DuplicateThreshold*). The output of the algorithm is 3 sets consisting of discovered relationships. If two datasets d_1 and d_2 are duplicates of each other they

[7]https://www.w3.org/TR/n-triples/

Algorithm 1: DatasetSimilarityMatching

```
Input: DLNTripleFiles, ProfileMetadata,
       ProfileThreshold, RelationThreshold, DuplicateThreshold
Output: Duplicates, Relationships, Outliers
begin
1   D ← (DLNTripleFiles, ProfileMetadata)
2   Duplicates, Relationships, Outliers ← {}
3   foreach d ∈ D do
4       P ← D \ {d}
5       foreach p ∈ P do
6           psimilarity ← AvgProfileSimilarity(d, p)
            if psimilarity > ProfileThreshold then
7               Sem ← parisSimilarity(d, p)
                foreach r ∈ Sem do
                    if s ∈ r > RelationThreshold then
8                       Relationships ← Relationships ∪ {r}
                    End If
                if ∀a1 ∈ Attributes(d), ∃a2 ∈
                  Attributes(p) ∧ (d, a1, p, a2, s) ∈ Sem ∧ s >
                  DuplicateThreshold then
9                   Duplicates ← Duplicates ∪ {(d, p)}
                End If
10      D ← D \ {d}
11  foreach d ∈ D do
        if ∄(d, a, d2, a2, s) ∈ Relationships then
12          Outliers ← Outliers ∪ {d}
        End If
13  return Duplicates, Relationships, Outliers
```

are added to the *Duplicates* set as a tuple (d_1, d_2), if they are not related to any other dataset then they are added to the *Outliers set* as a tuple of the dataset identifier (d_x) and if they are related to other datasets then the exact attributes from both datasets a_1 and a_2 with relationships (similarity measure s between 0 and 1) between them are added to the *Relationships set* as a tuple 'r' of (d_1, a_1, d_2, a_2, s).

The algorithm loops (Lines 3-10) on each dataset (and its accompanying profile) and compares it with each of the other datasets (in set 'P' from Line 4) based on the similarity of their data and schema profiles *psimilarity*. If the *psimilarity* is bigger than the assigned threshold in the input then the ontology similarity is computed within the inner-loop of Lines 5-9 which compares each dataset with each of the other datasets not checked before by the algorithm. This filtration If-statement (Line 7) is used to prevent non-necessary expensive comparisons with ontology alignment tools for datasets with disjoint profiles. To prevent any filtration in this step, we can set the *ProfileThreshold* to 0. The *psimilarity* is calculated as the average similarity of all data-profile and schema-profile metadata features in *ProfileMetadata* for both datasets in *AvgProfileSimilarity*(d_1, d_2). In Line 7 we compute the ontology similarity *parisSimilarity* [6] between each attribute of the dataset and attributes of the other datasets not checked by the algorithm before (we guarantee not double checking datasets by removing them from the comparison list of 'D' at the end of the loop in line 10). The set *Sem* in line 7 contains relationships tuples 'r'. In Line 9, if the all attributes between both datasets have relationships with similarity exceeding the *DuplicateThreshold*, then we add the datasets to the *Duplicates* set. In lines 11-12, if the dataset is not related to any other dataset (i.e. does not have any member tuple in the *Relationships* set), then we add it to the *Outliers* set in line 12. To make the algorithm more efficient we take samples of instances for comparison in the N-Triples of each dataset element of 'D'. The worst-case complexity of the algorithm is given in Equation 1.

VI. Experiments and Results

In this section, we describe the results of executing the prototype on the OpenML DL. To compare the automated approach and algorithm with the manual approach, we conduct an experiment with OpenML data. Our goal is to test the feasibility and effectiveness of our automated approach as compared to manual human checks. For the sample data of 15 datasets related to different domains as described in Table I, we present these data to 5 human-experts to analyse the relationships (like those listed in Table II) and then compare this to our automated approach. The human participants consisted of postgraduate pharmacists representing data enthusiasts. We have also independently analysed the datasets in 6 hours and have created a gold-standard of relationships, duplicates, and outliers for evaluating the manual and automatic approaches against. Such relationships detection includes analysis of two main types of attributes described below:

- **Numeric attributes:** Those include attributes represented as integers or real numbered values. They have a data profile involving statistical value distributions like mean, min, max, and standard deviations. An example would be the attributes in row no. 5 in Table II showing the continuous numeric value of the weight of cars in kilograms (e.g., 3000).
- **Nominal and String attributes:** Those include attributes having discrete values of nominal numbers or strings of characters. Their data profile mainly involves frequency distributions of their distinct values. An example would be the attributes in row no. 4 in Table II showing the name of the car models in the dataset in character strings. For example, the strings "volkswagen_type_3" and "Volkswagen". Although the values are represented in different strings of characters, they still hold the same information about *Volkswagen* cars and should be detected in the experiments as similar values.

For the automated CM4DL implementation, the thresholds used with Algorithm 1 were *0.5 or 0.0* for *ProfileThreshold*, *0.5* for *RelationThreshold* and *0.75* for *DuplicatesThreshold*. All experiments were executed on an i7-5500U Quad-core CPU, 8GB of memory and 64-Bit Windows 7 machine. We examine using the following alternatives:

- **Different sampling sizes:** we execute random sampling on the data instances to speed-up the ontology alignment task. We test using samples of sizes 100, 500, and 700 instances.
- **Different iteration counts until convergence:** In order to align the ontologies, the techniques used are usually iterative in nature and require multiple iterations until convergence [6]. The iterative nature allows for refinement of the matching results [11]. We test different number of iterations until we stop the alignment task. We test using 3,5,7, and 10 iterations.
- **Different similarity detection approaches:** We test two alternative approaches for similarity detection between attributes: the identity-based and the shingling-based matching.

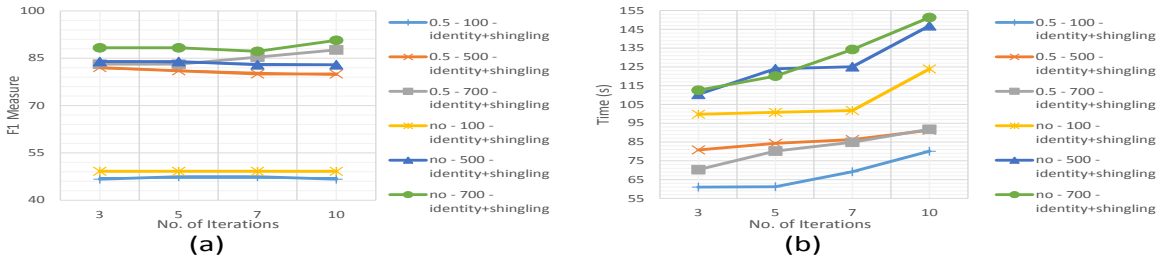

Fig. 4. Performance analysis of CM4DL in the OpenML experiments

TABLE III
RESULTS OF MANUAL ANNOTATION

Participant	Time Taken	Precision	Recall	F1
1	0.66 hours	91.3	55.3	68.9
2	4 hours	66.0	92.1	76.9
3	3 hours	20.5	42.1	27.6
4	2.66 hours	28.8	60.5	39.0
5	1.5 hours	80.8	55.3	65.6

We also combine both approaches to detect relationships by running them both on the data and merging the output.

- **Different profile similarity thresholds:** We examine using the average profile similarity between datasets as a filtering technique to eliminate comparisons using ontology alignment. This involves eliminating datasets having a profile similarity below a threshold. We test two thresholds: 0.5 and 0. For the later threshold, it means we do not filter any comparisons.

We compare the overall standard precision, recall and F1 measures [20] for the relationships detected. For the human-experts, their results are summarized in Table III. As can be seen, it takes considerable effort and time to manually compare the datasets. It took on average more than 2 hours and up to 4 hours to annotate the datasets by a human. The precision average is also considerably low at 57.5% (with a min of 20.5% and max of 91.3%). For the recall, it was also at a low average of 61.1%. The overall F1 mean is at 55.6% which shows a need for improvement by automated techniques.

The graphs in Figure 4 show the assessment of the F1 measure and computational efficiency (timing) of executing the automated algorithm on the experiment data. The time durations include the following: loading the datasets along with the JSON metadata to the triplestore, the tasks of the content metadata middleware in parsing the data and converting them into RDF N-Triples, and the ontology alignment execution time between all the datasets in the experimental setting. We test and compare for different number of iterations for the ontology alignment and matching execution, different sampling and different threshold of data profile similarity filtration. Graph (a) shows the F1 for the combined identity-similarity and shingling-similarity instance-matching techniques, and graph (b) shows the corresponding execution time of the Algorithm 1. Each line in the graphs represent the following as indicated in the legend: *ProfileThreshold* for Algorithm 1-sampling size-similarity technique. *ProfileThreshold* was tested at 0.5 and without any limit (as indicated by 'no').

From the graphs in Figure 4, it can be seen that the automatic approach yields good F1 scores between 82% and 91% for sample sizes between 500 and 700 instances. Generally, sampling negatively impacts the F1 score of the algorithm, however it has a bigger effect on smaller sample sizes like 100 instances which yielded F1 between 46% and 50%. Filtering the data profiles before comparisons proved to be effective in improving computational times while not severely impacting the F1 score. For a sample of 700 instances, we can still achieve 87% F1 (just 3% downgrading from comparing all datasets) while considerably saving computation time from 151s to 92s. It was noted, as expected, that more iterations of the ontology alignment algorithm yields more time for computations. However, there is no big downgrades from using less iterations, while it can considerably save processing time.

We only demonstrate the results from the combined approach in the graphs of Figure 4. However, it must be noted that the identity-similarity matching outperformed the shingling-similarity matching in all experiments. Shingling had an F1 score between 35% and 49% while taking more computation time between 63s and 82s for no filtering and 40s to 50s for filtering the data profiles. On the other hand, identity had an F1 score between 86% and 89% for sample sizes between 500 and 700 instances. For 100 instances samples, the effectiveness deteriorated sharply between 50% and 55%.

VII. DISCUSSION

As can be seen in the results, the automated techniques outperforms human subjects in both effectiveness (in terms of F1 measure) and efficiency (in terms of time for computation). By interviewing the human subjects, it was noted that they mainly focus on analysing nominal attributes without delving into numerical attributes analysis. In some subjects, there were almost no relations made within numeric attributes in the whole exercise, although they were instructed to do so. The automated techniques are more adept at comparing numeric features. On the flip side, humans are good at analysing nominal attributes as they can understand the semantic meanings behind them, which is difficult for a machine, e.g., for relationship no. 2 in Table II, it is obvious for humans that a year '1975' is similar to '75' but the automated algorithms considering shingling-sizes of 3 or 4 can not easily detect this relationships (as was experienced in our experiments). Also, the general feedback from human subjects expresses that they feel reluctant when making correlations between data. Such data enthusiasts simply do not want to spend considerable time to take the same systematic approach as a machine.

The results show that sampling can considerably minimize the algorithm's execution duration while not severely im-

pacting the performance of the algorithm. It was observed, as expected, that smaller sample sizes negatively impact effectiveness measured by F1. However, this impact becomes sharp with very small sample sizes only. For slight sampling variations in larger sample sizes, the F1 measure is not severely impacted. The results indicate that higher number of iterations for the alignment algorithms can yield better results, however, a more optimal number of less iterations can be selected without severely impacting the effectiveness of the algorithms. The efficiency gains from less iterations and more sampling can save considerable time. Sampling the data profiles from the datasets before comparison using ontology alignment techniques also saves sufficient computational time while not having considerable negative impact on the F1 score.

Duplicates were overall really well detected in all experiments applying the identity-based matching algorithm. Relationships in numeric attributes were better detected using identity-based matching, and for string attributes they were better detected using shingling-based matching. Combining both matching approaches led to the highest overall F1 score.

Shingling matching techniques generally have a low recall and precision but are good in detecting approximate relationships for string-based attributes. Shingling adds considerable errors especially in numeric attributes which reduces the precision. Identity-based techniques have high recall and precision but miss the cases of quasi-similarity string matching. Identity techniques have a high rate of recall and can easily detect duplicated datasets. It is therefore advisable to use multiple techniques in ontology alignment between datasets to improve the effectiveness in detecting the relationships between them.

The automated end-to-end process saves the huge manual effort required to analyse the datasets and annotating the metadata to the datasets. The automation results in some tens of seconds for metadata extraction and management instead of the multiple hours required by manual human inspection.

VIII. Conclusion and Future Work

We have presented our content metadata management framework which facilitates alignment in DL. We have demonstrated our approach within the OpenML DL environment. Our experiments shows the feasibility of our automatic approach in detecting relationships between the datasets. The results show that filtering the datasets for comparison, using sampling techniques, and using different ontology matching techniques can improve the efficiency of the approach while still achieving good effectiveness. We have also demonstrated the types of content metadata to collect for: schema, data profiles, and information profiles. This content metadata was used in a structured process to detect relationships between datasets, in order to facilitate the navigation and analysis of the DL.

For the future, we will examine the utilization of different supervised learning techniques to find the optimum similarity thresholds and weightings of the similarity measures to use in our algorithm. We will also investigate how to dynamically select the sample size based on a measure of heterogeneity of the datasets being compared together. We also acknowledge that we can improve the efficiency of the algorithm by creating a 3^{rd} reference integration ontology after each ingestion to decrease the number of comparisons by the algorithm to this single-integrated ontology. To improve the efficiency of our algorithm we are planning to parallelize the computations in a parallel-computing framework like MapReduce.

Acknowledgement. This research has been funded by the European Commission through the Erasmus Mundus Joint Doctorate (IT4BI-DC).

References

[1] I. Terrizzano, P. Schwarz, M. Roth, and J. E. Colino, "Data Wrangling: The Challenging Journey from the Wild to the Lake," in *7th Biennial Conference on Innovative Data Systems Research CIDR'15*, 2015.

[2] K. Morton, et al., "Support the Data Enthusiast : Challenges for Next-Generation Data-Analysis Systems," *Proceedings of the VLDB Endowment*, vol. 7, no. 6, pp. 453–456, 2014.

[3] J. Varga, et al., "Towards Next Generation BI Systems : The Analytical Metadata Challenge," *Data Warehousing and Knowledge Discovery - Lecture Notes in Computer Science*, vol. 8646, pp. 89–101, 2014.

[4] H. Alrehamy and C. Walker, "Personal Data Lake With Data Gravity Pull," in *IEEE Fifth International Conference on Big Data and Cloud Computing (BDCloud)*, 2015, pp. 160–167.

[5] P. a. Bernstein, J. Madhavan, and E. Rahm, "Generic Schema Matching , Ten Years Later," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 695–701, 2011.

[6] F. M. Suchanek, S. Abiteboul, and P. Senellart, "PARIS : Probabilistic Alignment of Relations , Instances , and Schema," *Proceedings of the VLDB Endowment*, vol. 5, no. 3, pp. 157–168, 2011.

[7] F. Naumann, "Data profiling revisited," *ACM SIGMOD Record*, vol. 42, no. 4, pp. 40–49, 2014.

[8] R. Hauch, A. Miller, and R. Cardwell, "Information Intelligence : Metadata for Information Discovery , Access , and Integration," in *ACM SIGMOD international conference*, 2005, pp. 793–798.

[9] V. Santos, F. A. Baião, and A. Tanaka, "An architecture to support information sources discovery through semantic search," in *IEEE International Conference on IRI*, 2011, pp. 276–282.

[10] Z. Abedjan, L. Golab, and F. Naumann, "Profiling relational data: a survey," *The VLDB Journal*, vol. 24, no. 4, pp. 557–581, 2015.

[11] S. Lacoste-Julien, et al., "SiGMa: Simple Greedy Matching for Aligning Large Knowledge Bases," in *Proceedings of the 19th ACM SIGKDD international conference*, 2013, p. 572.

[12] M. Piernik, D. Brzezinski, and T. Morzy, "Clustering XML documents by patterns," *Knowledge and Information Systems*, vol. 46, no. 1, pp. 185–212, 2016.

[13] K. Murthy, et al., "Exploiting Evidence from Unstructured Data to Enhance Master Data Management," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 1862–1873, 2012.

[14] M. Interlandi, K. Shah, S. D. Tetali, M. A. Gulzar, S. Yoo, M. Kim, T. Millstein, and T. Condie, "Titian: Data Provenance Support in Spark," *Proc. VLDB Endow.*, vol. 9, no. 3, pp. 216–227, 2015.

[15] S. Bykau, et al., "Bridging the Gap between Heterogeneous and Semantically Diverse Content of Different Disciplines," in *IEEE Workshops on DEXA*, 2010, pp. 305–309.

[16] R. Touma, O. Romero, and P. Jovanovic, "Supporting Data Integration Tasks with Semi-Automatic Ontology Construction," in *ACM Workshop on DOLAP*, 2015, pp. 89–98.

[17] S. Moawed, et al., "A Latent Semantic Indexing-Based Approach to Determine Similar Clusters in Large-scale," *New Trends in Databases and Information Systems*, pp. 267–276, 2014.

[18] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "OpenML: networked science in machine learning," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 2, pp. 49–60, 2014.

[19] R. Steorts, S. Ventura, M. Sadinle, and S. Fienberg, "A Comparison of Blocking Methods for Record Linkage," in *International Conference on Privacy in Statistical Databases*, 2014, pp. 253–268.

[20] C. D. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*. Cambridge University Press, 2009.