



HAL
open science

PROCLAIM: An Unsupervised Approach to Discover Domain-Specific Attribute Matchings from Heterogeneous Sources

Molood Arman, Sylvain Wlodarczyk, Nacéra Bennacer Seghouani, Francesca
Bugiotti

► **To cite this version:**

Molood Arman, Sylvain Wlodarczyk, Nacéra Bennacer Seghouani, Francesca Bugiotti. PROCLAIM: An Unsupervised Approach to Discover Domain-Specific Attribute Matchings from Heterogeneous Sources. CAiSE'20, 32nd International Conference on Advanced Information Systems Engineering, Jun 2020, Grenoble, France. hal-02863603

HAL Id: hal-02863603



<https://hal.science/hal-02863603>

Submitted on 10 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PROCLAIM: An Unsupervised Approach to Discover Domain-Specific Attribute Matchings from Heterogeneous Sources

Molood Arman^{1,2} , Sylvain Wlodarczyk¹ , Nacéra Bennacer Seghouani² , and
Francesca Bugiotti² 

¹ Services Pétroliers Schlumberger, 34000, Montpellier, France

² Université Paris-Saclay, CNRS, Laboratoire de Recherche en Informatique, 91405, Orsay, France

{marman2, swlodarczyk}@slb.com

{nacera.seghouani, francesca.bugiotti}@lri.fr

Abstract. Schema matching is a critical problem in many applications where the main goal is to match attributes coming from heterogeneous sources. In this paper, we propose PROCLAIM (PROfile-based Cluster-Labeling for Attribute Matching), an automatic, unsupervised clustering-based approach to match attributes of a large number of heterogeneous sources. We define the concept of attribute profile to characterize the main properties of an attribute using: (i) the statistical distribution and the dimension of the attribute’s values, (ii) the name and textual descriptions related to the attribute. The attribute matchings produced by PROCLAIM give the best representation of heterogeneous sources thanks to the cluster-labeling function we defined. We evaluate PROCLAIM on 45,000 different data sources coming from oil and gas authority open data website³. The results we obtain are promising and validate our approach.

1 Introduction

During the last years, the availability of multiple and heterogeneous data sources has given new perspectives to the schema matching problem which is a fundamental step for data integration. A large number of research works exist in the literature, the main task in these approaches is to identify the correlation between the attributes using dataset values, semantic and syntactic rules to detect the correspondence between attributes during the schema matching process [1]. Most of the works on schema integration assumed a global (mediated) schema and then tried to find a solution for better matching on mostly a pairwise matching between the source schema and the mediated schema. In this context it is very difficult to define a global schema that matches all the attributes of a given domain [10]. Moreover, real world data is always noisy and for most of integration methods, data cleaning is needed. However, in terms of big data, data cleaning is

³ The data is published under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

expensive and time consuming. In this paper, we develop an heuristic method which can deal with real world and massive data.

In this paper, we present PROCLAIM (PROfile-based Cluster-Labeling for Attribute Matching), an unsupervised method for matching attributes coming from a large number and heterogeneous sources in a specific domain. Our results show that PROCLAIM is an effective fully automatic method to discover a set of meaningful vocabularies which are the backbone of the definition of a specific domain. PROCLAIM defines the concept of attribute profile by taking into account the data type using: (i) the statistical distribution and the dimension of the attribute's values, and (ii) the name and textual descriptions of the attribute. These properties give a unified representation to each attribute. The cluster-labeling function takes as input these properties to automatically assign a set of labels to a high number of attributes.

The paper is organized as follows: Section 2 reviews the related studies on schema matching and the available tools. Section 3 presents a brief overview of PROCLAIM. Sections 4, 5 and 6 detail each building block of PROCLAIM. Section 7 illustrates the results of our experiments in two different domains. Finally, Section 8 draws some future steps.

2 Related Work

Knowledge Base (KB) construction is a recurrent problem in industry and research and includes problems of data extraction, cleaning, and integration [6]. A significant amount of work has been done in recent years for automatic construction of knowledge bases. However, the first step of KB construction, which is defining a global schema with the aim of populating the KB, still requires a manual effort [15]. Several previous researches were mainly focused on extracting data from unstructured data such as texts. Open Information Extraction systems are not concerned with the integration of extracted entities and their properties from different sources with unified names. Because of this limitation the resulting knowledge bases may the same entity represented multiple times with different names [15]. Other techniques, such as Biperpedia [9], use search engine query logs in addition to text to discover attributes. This process involves numerous trained classifiers and corresponding labeled training data. Most of the automatic KB construction systems were focused on retrieving facts and entities from unstructured datasets. To our knowledge, integrating the existing structured sources in the knowledge bases has not been considered in the process of constructing the KB automatically.

A large number of publications focused only on schema matching. In this context, schema matching identifies the correspondences between similar elements belonging to different schemas. IntelliLIGHT [8] is a system looking in large-scale structured data sets which aims to locate and retrieve needed data in a specific domain. It proposes a method which ranks the main data tables taking as output the ones having a higher score. PROCLAIM is a very different approach to the problem; instead of ranking the best available schema among different data sources, it provides a unified standard schema from all sources and generates a global schema for a domain automatically. UFO [11] is a data structure expressing various representations of the same concept as a data object and is capable of recognizing and mapping such objects in different data sources automatically. The

WebTable system [3] is a search engine that ranks tables scraped from the web. In this approach, AcsDB is introduced as a database which contains a corpus of statistics on schema elements that is used to compute the probability of an attribute (the number of schemas containing the attribute divided by the total number of schemas) and the probability of an attribute conditioned on another attribute. WebTable autocompletes a schema (suggest additional related attributes for a given set of attributes) by using the probability of pair attributes in different schema to provide additional synonyms. In contrast, PROCLAIM focuses on all characteristics of all attributes to find the similar attributes in the provided schemas. The main goal of PROCLAIM is discovering the most complete global schema over the existing schemas in a domain.

3 PROCLAIM Overview

Schema matching aims at discovering semantic correspondences of attributes of schemas across heterogeneous sources. Our goal is to get a global attribute schema for all the independently developed schemas of the same domain which can be formalized as follows.

Given a set of schemas $\mathcal{S}=\{S_1, S_2, \dots, S_n\}$ and the set of all attributes $\mathcal{A}=\{A_1, A_2, \dots, A_n\}$ belonging to these schemas, each A_i contains the whole set of attributes (a_1, \dots, a_m) used in the schema S_i . Let us consider a single *schema* (S) and its set of attributes (A) ($a_i \in A$ where $i \in [1 : m]$). *Schema matching* selects sets of *n*-ary *mapping attributes* which together define similar groups of attributes (G_i), as illustrated in Example 1. All attributes are trivially a group by themselves. A *label* (l) can identify in the best way the essence of a semantic group of attributes. A *labeling function* $f_L(G)$ indicates the required process to define the label ($f_L : G \rightarrow L$), where L is a set of labels ($l_i \in L; 0 \leq i \leq m$) and G is a set of similar groups of attributes ($G_i \in G$). The set of labels (L) identifies the elements of a global schema for the given set of schemas (\mathcal{S}), this resulting schema is also the mediated or target schema.

Example 1. Consider three schemas as set of attributes about rental cars descriptions:

$S_1 = \{\text{Fuel_Type, Location, Mileage, Name, Price, Year, Transmissio}\}$

$S_2 = \{\text{Country, Disp., HP, Mileage, Price, Type}\}$

$S_3 = \{\text{fuel_type, maker, manufacture_year, mileage, model, price_eur, transmission}\}$

Also, consider the following attribute matches among the schemas:

$G_1 = \{\text{Fuel_Type, fuel_type, fuel, fuelType}\}$

$G_2 = \{\text{Location, Country, city, county_name, state_name}\}$

$G_3 = \{\text{Name, maker, brand}\}$

Consider these labels $\{\text{Location, Brand, Fuel, Idots}\}$ extracted from the data sets. These labels will be assigned to each group of attribute (i.e. as follows: $l_1 = \text{Fuel}$, $l_2 = \text{Location}$, $l_3 = \text{Brand}$) and will define the names of the attributes of the global schema for a specific domain: $L = \{\text{Fuel, Location, Brand, \dots}\}$.

The main question addressed in this research is how to define an automatic process that discovers a set of labels which can effectively represent a global attribute schema for a specific domain. The PROCLAIM method is proposed as an answer to this question. PROCLAIM is a new approach which enables the automatic holistic schema matching

which leads to construction of a global attribute schema for a specific domain. Let us illustrate the procedure, by following the main steps it involves, with the help of Figure 1:

1. a set of heterogeneous sources with different schema (\mathbf{S}) is provided as input;
2. the data from all sources are stored in columnar format storage;
3. the data type of each attribute is identified and data with the same data type are stored in the same set (\mathbf{S}_{d_k});
4. an attribute profile is computed based on the specificity of each data type (\mathbf{S}_{d_k}). This profile for all kinds of attributes can contain at most four properties (statistics, description, unit, and name property). Then each profile of attributes can contain at most four properties. The assigned profile to each attribute will be converted to a numerical vector;
5. an automatic labelling process is defined to find all similar attributes and gives a unified name to each of them. This process includes two principal components: (1) finding the most similar attributes from different schemas, (2) giving an automatic label to each attribute by a defined labeling function (L_f). A density based clustering algorithm will be applied on the numerical profiles for finding the most similar attributes. Each profile vector represents a unique attribute;
6. the list of automatically computed labels will define a global attribute schema for a specific domain.

As explained in detail in the following sections, PROCLAIM can be applied on real-life noisy data. The method is designed to handle a large number of heterogeneous schemas and proposes a unified numerical profiling of information of any data type. The approach enables the usage of common machine learning algorithms such as clustering. Finally the automatic labeling and merging of clusters allow the definition of a global schema that represents the synthesis of the heterogeneous schemas.

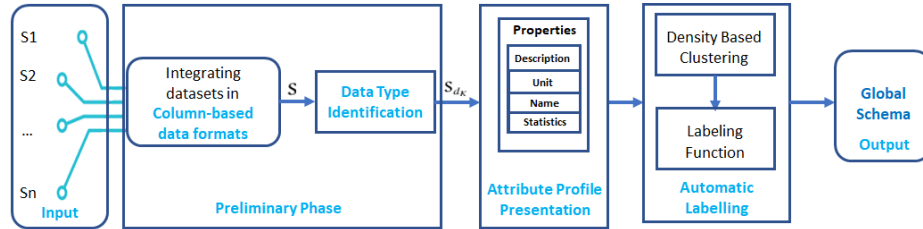


Fig. 1: The framework of PROCLAIM to discover a global schema

4 Preliminary Phase

Some of the building blocks of PROCLAIM can be considered as initial steps to prepare the original datasets. Two main steps are defined as the initial steps in the preliminary phase of PROCLAIM (1) targeting data into a columnar datastore, (2) identifying the data type.

4.1 Column-based data formats

Column-based data formats organize data in a set of tables. Each table contains a set of rows, and each row has a set of columns, each with a name and a value. Rows in a

table are not required to have the same attributes. Data access operations are usually over individual rows and show the best performances when retrieve only a subset of the attributes of a table, when data sets are sparse and contain lots of empty values [12]. Moreover column-based data formats process big datasets efficiently since provide large-scale parallelization and effective partitioning strategies. PROCLAIM for its calculation needs a tuple for each value of attributes showing the name of the attribute and its value. In this case, storing the data in columnar-based format is much efficient.

4.2 Data Type Identification

When the search space is large (the number of attributes or schemas are big), matching the complete input of schemas may require long execution times, and achieving high quality results may be difficult. One way to reduce the search space is to find the similar attributes within the same data types. The heterogeneous sources provide attributes in different data types. Since the type of the attributes may not be provided in the metadata of sources, we need to identify the types given the values. One main problem in this step is the fact that the original datasets are not clean. We have to consider the type based on the data type of the majority of the instances (values). Here, we just consider five data types but this set can be extended if it is necessary:

- NUMERICAL representing all attributes which its value just contain integer or float.
- CATEGORICAL containing all strings, characters and mix data type.
- DATE representing date and time such as datetime, timestamps and etc.
- RARE classifying attributes which they have less than 10 instances.
- UNIQUE referring to attributes with unique value (cardinality is equal to 1).

Formally, let d_k be the data type of an attribute a_i with probability $p \geq \text{threshold}$ (e.g. *threshold* 0.8) where $a_i \in \mathcal{A}$ and $d_k \in \{ \text{NUMERICAL, CATEGORICAL, DATE, RARE, UNIQUE} \}$ where $k \in [1,5]$. *Data Type function* (F_D) ($F_D : \mathcal{A} \rightarrow \{ \text{NUMERICAL, CATEGORICAL, DATE, RARE, UNIQUE} \}$; $F_D(a_i) = d_k$) pre-classified the attributes of the whole dataset into maximum five categories (\mathcal{S}_{d_k}) which contain attributes with the same data type.

5 Attribute Profile Representation

Once we have all the attributes belonging to the same data type (\mathcal{S}_g) we can group them to discover attributes coming from different schemas which contain the same information (e.g., {name, maker, brand} in our example). PROCLAIM performs clustering and labeling based on the computation of a similarity matrix of numerical profiles of attributes. Before applying our algorithm, we must convert an attribute to a numerical profile based on its data type. According to our representation any attribute is characterized by a maximum of four components according to the data type to which it belongs. These components are description, unit, name, and statistics. In this section we provide a description of each component of the profile and its contribution in the analysis of the attributes classified in any of the six data types introduced in the previous section. Notice that the RARE type attributes are ignored due to the impossibility to compute a valid statistic.

Description Property The majority of datasets have a descriptive part for the schema where the meaning of each attribute can be found. In other cases, the description is not provided but the used values belong to domain specific terms or abbreviations and this description can be retrieved, for example, using domain specific Wikis.

To create the description profile, first of all, we remove the stop-words and then we apply the stemming method over a bag of tokens. Then for each description, the stems and the occurrence of each term (in all the different descriptions for any specified attribute) is used to build the description profiles. Removing stop-words in a specific domain is necessary, since these words can appear in almost all descriptions and can cause false similarities (e.g., for the domain of cars, the words such as car, vehicle, automobile and etc, are the domain stop-words). We then transform the descriptions to categorical variables. Next, feature engineering is required to encode the different categories into a suitable numerical feature vector. One-hot encoding is a simple but efficient widely-used encoding method [4]. An example of converting categorical variables for some attributes to numerical values can be seen in Table 1.

Attribute	displac	volum	engin	cc	repres	kw	ccm
ENGINE_DISPLACEMENT	0	0	0	0	0	0	1
ENGINE_POWER	0	0	0	0	0	1	0
DISP.	1	0	1	0	1	0	0
ENGINE	1	1	1	1	0	0	0

Table 1: One-hot encoding for converting descriptions to numerical feature

Unit Property Dimensions and units are fundamental tools to explain the characterization of phenomena [13]. A dimension is a measure of a physical variable by fundamental quantities without numerical value, such as distance, time, mass, and temperature. However, a unit is a specific way to assign a measurement (with numerical value) to the dimension, e.g., a dimension is length, whereas meters or feet are relative units that describes length [13]. Dimensions and units are commonly confused, despite the fact that the solution to most problems must include units. The distribution of the same entity in different units can be shifted, but by consideration of the same dimension, the similarity of shifted distribution can be found. Also, attributes with units related to same dimension are related to each other through a conversion factor, such as Kelvin or Celsius which measures the dimension of temperature and they can convert to each other. Given a dataset, the related units can be found thanks to the descriptive part of the schema or taking into account also the instances (near the value or in a separated column). The units and their mapped dimensions of attributes can be extracted and recorded separately. In Table 2 we show dimensions and units characterizing some attributes of our running example. The dimension is also encoded using one-hot encoding approach.

Attribute	Unit	Dimension
ENGINE_DISPLACEMENT	CCM	VOLUME
ENGINE_POWER	KW	POWER
PRICE_EUR	EUR	PRICE
ENGINE	CC	VOLUME

Table 2: Some attributes with their units and associated dimension

Name Property The name of an attribute can also be useful for the analysis. Names often contain concatenated words and abbreviations. Thus, they first need to be normalized before they are used to construct a profile to compute linguistic similarities. First

tokenization is applied but it may not be enough; e.g for the name 'vehicleType', the name should be split into word 'vehicle' and 'Type'. In this regard, we compare all names of other attributes and see if one of them is part of the name string, this breakdown will be done.

Statistics Property The statistics profiles concern CATEGORICAL and NUMERICAL data types. PROCLAIM uses descriptive statistical analysis to produce a profile for each attribute which not only defines the characteristics of an attribute but also enables comparing the profiles to find similarities. In the following, we list the most important statistical measurements regarding NUMERICAL and CATEGORICAL data types.

– NUMERICAL data type:

For the NUMERICAL data type, there are several measures that can be studied. The domain under analysis and the characteristics of analyzed data will help us to select the significant ones. these measures can be variability or dispersion of distribution of values per each attribute, symmetry of the distribution, the number of instances (cardinality) and central tendency.

– CATEGORICAL data type

For the CATEGORICAL data type, the considered statistics profile contains the top most frequent values among all instances of one attribute. This set of top most frequent instances can design a pattern for an attribute.

Since other components of attribute profiles are encoded using one-hot encoding approach we decided to apply the same method to the statistics profile. First log transform will normalize the distribution with left or right skewness, then the distribution is presented into categorical scale using binning and finally encoded. We obviously lose the numerical nature of the statistics but we can merge easily this vector with the other vectors without a normalization issue.

Attribute	5%	25%	50%	75%	95%	Count
DISP.	90.9	113.75	144.5	180.0	302.0	32
ENGINE	993.0	1198.00	1497.0	1995.0	2982.0	101
ENGINE_DISPLACEMENT	1124.0	1400.00	1600.0	1968.0	2967.0	158
ENGINE_POWER	44.0	65.00	80.0	103.0	161.5	114

Table 4 a: Statistics Profile

Attribute	5%	25%	50%	75%	95%	Count
DISP.	5	5	5	5	6	3
ENGINE	7	7	7	8	8	5
ENGINE_DISPLACEMENT	7	7	7	8	8	5
ENGINE_POWER	4	4	4	5	5	5

Table 4 b: Normalized Statistics Profile

Example 2. In Table 4 (a) we present the statistics profile for four numerical attributes. As a result of this analysis, we can see that the 'Engine' and 'Engine Displacement' have the same normalized distribution. Normalized data with log transformation is shown in Table 4 (b).

For each attribute of the dataset, we compute the global profile which is made of four properties we described in this section. Each profile is built by considering the type of attribute and the global profile is finally converted into a numerical vector.

We finally produce a dataset that is made of a collection of vectors that will be the input for the next steps of the computation.

For each of the four properties, we propose a weighting factor on the properties that is adjusted according to the data type of the attribute. For example, for numerical and categorical variables, the attribute name can be ignored because this information is uncertain and the distribution of the values is very important.

6 Attribute Labeling

The attribute labeling is a three step process that (1) performs attribute clustering, (2) assigns a label to each cluster, and (3) merges clusters having the same label. Step 3 creates each single attribute of the global schema. In this section, we are going to detail each step of the process.

6.1 Clustering

The calibrated numerical vectors produced as described in Section 5 allow us to apply clustering to find similar groups of attributes ($G_i \in G$). PROCLAIM uses a density-based clustering method. Density-based clusters are connected, dense areas in the data space separated from each other by low density areas. Density-based clustering can be considered as a non-parametric approach, since this method makes no assumptions about the number of clusters or their distribution [5]. In higher-dimensional space the assumption of a certain number of clusters of a given distribution is very strong and may often be violated. However, other parameters should be identified, e.g., a density threshold that is the minimum number of points (MinPts) and the radius of a neighborhood (ϵ) in the case of DBSCAN [7] and OPTICS [2]. Sparse areas as opposed to high density areas are considered as outliers (noise). This results in having points in the sparse areas that are not assigned to any cluster since in general each outlier can be considered as one cluster containing just one element. As a result, 1) It is not necessary to specify the number of clusters; 2) It is not necessary that all the points belong to at least one cluster.

OPTICS [2] (Ordering PoinTs to Identify the Clustering Structure) and the aforementioned DBSCAN are two popular density based clustering algorithms. Despite all the similarities in the core concept of both algorithms, they have fundamental differences [2]. PROCLAIM uses OPTICS. In PROCLAIM, we want to reduce the chain of core profile effect [2] in order to have small clusters with very similar profiles; hence, we set a very small value (e.g 3) for the MinPts input of OPTICS. We will then compute many clusters and have many outliers. To reduce the number of outliers we run OPTICS, a second time, again with a small value for the MinPts parameter only on the profiles that were considered as outliers. The clusters computed during the second step will be added to the clusters computed at the first step. With these two iterations, we increase the number of clusters and reduce the outliers.

6.2 Labeling Function

The labels for each cluster will be created by using the descriptions and names of all elements in each cluster. The stop words will be removed using the common linguistic stop words and the domain specific ones. The idea is to select the most frequent words, bigram, and trigram terms appearing in the description and name of each attribute of the cluster. Then, the most frequent term will be the label of the cluster as shown in Example 3.

Example 3. Consider $C_1 = \{ENGINE, DISP.\}$ as a cluster computed using the two-steps OPTICS algorithm. The descriptions gathered per each attribute are:
Descr_Engine = ' The displacement volume of the engine in CC.'
Descr_Disp. = ' : Represents the engine displacement of the car'
 The name profile of attributes can also be added to the descriptions: *Descr_names* = {*engine, disp.*}.
 Furthermore, after removing the stop words, the following bag of words for each description will be generated:

BOW_Engine = {displacement : 1, volume : 1, engine : 1, cc : 1}

BOW_Disp. = {represents : 1, engine : 1, displacement : 1, car : 1}

BOW_names. = {engine : 1, disp : 1}

Moreover, we create a holistic bag of words by merging all the terms together associated with their total number of occurrences as follows:

BOW_total = {engine : 3, displacement : 2, volume : 1, cc : 1, represents : 1}

By selecting the most represented term, we may produce some meaningless labels such as "displacement engine" rather than "engine displacement". To tackle this problem, we need to create a domain specific corpus and extract from it the bigrams and trigrams associated with the respective number of occurrences. This will be used to adjust and validate the labels.

Consider a created corpus in the cars domain which includes resources of glossaries, dictionaries, wikis and etc. which can easily be gathered online⁴. Now, all combinations of the highest frequency words from *BOW_total* will be considered to create the bigrams and trigrams which already exist in this domain (the meaningful N-grams) with respect to terms frequency in the corpus. The bigrams and trigrams selected will create a valid bag of terms. We will also add the most frequent word appearing in the corpus to this valid bag of terms. From Example 3 we have: *Bag_of_terms* = {engine displacement : 2, displacement volume : 1, engine : 3}. To get the selected label, we take from the bag of terms the term with the maximum number of occurrences with the priority first to the trigrams then bigrams and finally words.

The selected label for the cluster $C_1 = \{ENGINE, DISP.\}$ is ***engine displacement*** even if the number of occurrences of ***engine*** is higher.

After labeling each cluster we can finally merge the clusters with the same label or labels that are synonyms (Example 4).

Example 4. Consider $C_2 = \{ENGINE_DIS PLACEMENT, ENGINE_POWER\}$ as another cluster computed using the two-step OPTICS algorithm. The bag of words retrieved from related descriptions for these attributes are:

BOW_Engine_Displacement = {ccm : 1}

BOW_Engine_Power = {kw : 1}

BOW_names = {engine : 2, displacement : 1, power : 1}

As final result the output is:

Bag_of_terms = {engine displacement : 2, engine power : 1, engine : 3}

⁴ Data from: <https://www.kaggle.com/>

The computed label is again *engine displacement* which means that this cluster can be merged with cluster C_1 of the example 3. Then the new cluster contains the following attribute {ENGINE, DISP., ENGINE_DISPLACEMENT, ENGINE_POWER}.

All the merged and labelled clusters generate a global schema for a specific domain. The label of different clusters in different data types can be the same which enables us to integrate the attributes together even if their data types were assigned wrongly in Section 4.2. PROCLAIM helps to integrate the data from different sources and also creates a general schema which can help for integration or new sources or to populate a knowledge base in the specific domain.

7 Experiment Results

In this section, we provide the experimental results on two datasets: one of them is our ongoing cars example and the second is from the oil and gas domain. The code of the experiment is implemented in Python 3.6.7. Parquet [14], a columnar datastore is used to store original datasets. Parquet is a free and open-source optimized column-oriented data storage developed on the Apache Hadoop ecosystem. To the best of our knowledge, there are no benchmark labeled datasets for comparing our results with another method. Therefore, for the car example, we have collected data from Kaggle challenges. For the Oil and Gas example, we use a large dataset.

Data set	Kaggle Challenge Name	#Attributes	#Descriptions	#Units	#Source records
S ₁	Used Cars Price Prediction	13	11	4	1000
S ₂	cars data	8	7	0	600
S ₃	personal cars classified	16	11	4	1000
S ₄	Craigslist Cars EDA	26	24	0	1000
S ₅	Used cars database	20	12	1	1000
Sum	Car_Kaggle	70	65	9	4600

Table 5: Car_Kaggle Data set Information as the input for PROCLAIM

Car_Kaggle The Car_Kaggle dataset was gathered from five different sources (S₁, . . . , S₅) about cars from different Kaggle challenges⁴. The global Car_Kaggle dataset, after merging different sources contain 78 original attributes: 70 of them have different names; 65 out of 70 attributes contain descriptions and just 9 out of 70 attributes have the provided unit. In Table 5, we provide the details of each schema. As first step, we run data type identification in order to discover the type of each attribute. Data for this dataset can be split in four different types and, as we can see the RARE data type is not present. Unique attributes are discarded (6 attributes) and we compute the profile for the 64 remaining attributes (25 NUMERICAL, 35 CATEGORICAL and 4 DATE attributes) and automatically assign label to each attribute. To be able to evaluate PROCLAIM, we manually labeled all the attributes. A subset of PROCLAIM labels and manual labels can be seen in Table 2a. To evaluate the quality of PROCLAIM labels, we used three metrics: *precision*, *recall* and *F-measure*. Precision is defined as the percentage of the correct labels. We compared manual labels with PROCLAIM labels. If the pair (PROCLAIM LABEL, MANUALLY ANNOTATED LABEL) matches, the label is considered as valid, as it can be seen in Table 6 (a). Recall is the ratio of attributes with correct labels to all attributes (with or without labels). F-measure which is the harmonic mean of the

Attributes	PROCLAIM labels	Annotated labels	Match
PRICE_EUR	price converted	price	1
PRICE	price converted	price	1
POWERPS	power	power	1
HP	power	power	1
WEIGHT	weight	weight	1
POSTALCODE	weight	address	0

Table 6 a: Labeling for Car_Kaggle

Data type	Precision	Recall	F-measure
NUMERICAL	85.7	85.7	85.7
CATEGORICAL	73.0	58.8	64.2
DATE	100	100	1
Overall	82.5	72.7	77.3

Table 6 b: PROCLAIM Evaluation

precision and recall. This result is shown in Table 6 (b). These measures were calculated separately for each set of attributes (of each data type) and finally for the whole set of attributes. As it can be seen in Table 6 (b), precision is showing a good quality of labels but since the number of attributes and sources are not big, we expected not very high recall, but still this recall is promising for the schema matching problem which in this research is not the main concern. The main goal is to have the labels with high quality.

Oil_NorthSea dataset The North Sea Oil and Gas (Oil_NorthSea) dataset were gathered from OGA³ (The Oil and Gas Authority Open Data) website which contains 43997 different sources with a total of 5260 attributes. 4713 of them have different names. The description is available for 3481 attributes and unit is provided for 1668 over 4713 attributes. We apply the same approach as described in section 7. The number of different identified types of attributes is: 638 NUMERICAL, 631 CATEGORICAL, 46 DATE, 574 RARE and 2824 UNIQUE attributes. Since the number of attributes is too big to be entirely manually annotated, we asked domain experts to label random set of attributes (20 labels for NUMERICAL and CATEGORICAL attributes and all labels for DATE attributes - the number of DATE attributes are less than 50). We cannot calculate recall and f-measure here, since the manual labels are just provided for a subset of random labels. However, precision is calculated for these subsets for different experiments. Experiments are done for different profiles for each group of same data type attributes and the result is shown in Table 7. Cover data ratio measures the percentage of labeled attributes. The Covered_data ratio is showing a high percentage of considered attributes to discover the global schema. As can be seen, the precision of clusters for NUMERICAL, CATEGORICAL, and DATE data type is over 90% which is a promising result. The global schema created from the Oil_NorthSea dataset contains 247 labeled attributes which covers 86% of the 1315 original attributes belong to the NUMERICAL, CATEGORICAL, and DATE data types.

Data type	Profile	#Unlabeled Attr.	#Labeled Attr.	#Labels	Precision (%)	Covered_data (%)
Numerical	Stat.	84	554	107	58.1	86.8
	Descr.	122	516	112	93.25	80.9
	[Stat., Descr.]	53	585	128	86.4	91.6
	[Stat., Descr., Unit]	60	578	110	90.1	90.5
Categorical	Stat.	135	496	102	70.5	78.6
	Descr.	203	428	100	94.9	67.8
	[Stat., Descr.]	129	502	126	86.2	79.5
	[Stat., Descr., Unit]	121	510	130	92.1	80.8
Date	Descr.	16	30	3	100	65.2
	[Descr., Name]	5	41	7	94.3	89.1
	[Descr., Unit, Name ¹]	5	41	7	94.3	89.1
Total	[full profile]	186	1129	247	92.2	85.9

¹Unit is not available for Date attributes

Table 7: Experiments results for different profiles subset

8 Conclusion and Future works

Compared to the huge work on pairwise schema matching, research on holistic schema matching for more than two sources is still at an early stage. PROCLAIM is an efficient and effective way for schema matching and provides a consistent domain-specific attribute schema. Experiments show that thanks to our approach we can gather automatically more than 80% of vocabularies related to a domain and populate the knowledge bases with corresponding attributes from heterogeneous sources. In future work, our approach can be extended for handling new attributes from new sources and for enriching the set of labels by adding similar words from different thesauri and dictionaries.

References

1. Alwan, A.A., Nordin, A., Alzeber, M., Abualkishik, A.Z.: A survey of schema matching research using database schemas and instances. *IJACSA* **8**(10) (2017)
2. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: ordering points to identify the clustering structure. In: *ACM Sigmod record*. vol. 28, pp. 49–60. ACM (1999)
3. Cafarella, M.J., Halevy, A., Wang, D.Z., Wu, E., Zhang, Y.: Webtuples: exploring the power of tables on the web. *Proceedings of the VLDB Endowment* **1**(1), 538–549 (2008)
4. Cerda, P., Varoquaux, G., Kégl, B.: Similarity encoding for learning with dirty categorical variables. *Machine Learning* **107**(8-10), 1477–1494 (2018)
5. Charu, C.A., Chandan, K.R.: *Data clustering: algorithms and applications* (2013)
6. De Sa, C., Ratner, A., Ré, C., Shin, J., Wang, F., Wu, S., Zhang, C.: Deepdive: Declarative knowledge base construction. *ACM SIGMOD Record* **45**(1), 60–67 (2016)
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. (1996)
8. Gubanov, M., Priya, M., Podkorytov, M.: *Intellilight: A flashlight for large-scale dark structured data* (2017)
9. Gupta, R., Halevy, A., Wang, X., Whang, S.E., Wu, F.: Biperpedia: An ontology for search applications. *Proceedings of the VLDB Endowment* **7**(7), 505–516 (2014)
10. Jiang, S., Liang, J., Xiao, Y., Tang, H., Huang, H., Tan, J.: Towards the completion of a domain-specific knowledge base with emerging query terms. In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. pp. 1430–1441. IEEE (2019)
11. Kola, A., More, H., Soderman, S., Gubanov, M.: Generating unified famous objects (ufos) from the classified object tables. In: *IEEE Big Data*. pp. 4771–4773. IEEE (2017)
12. NEXLA: An introduction to big data formats understanding avro, parquet, and orc. In: *NEXLA White paper*. pp. 1–12 (2018)
13. Rubenstein, D., Yin, W., Frame, M.D.: *Biofluid mechanics: an introduction to fluid mechanics, macrocirculation, and microcirculation*. Academic Press (2015)
14. Vohra, D.: Apache parquet. In: *Practical Hadoop Ecosystem*, pp. 325–335. Springer (2016)
15. Winn, J., Guiver, J., Webster, S., Zaykov, Y., Kukla, M., Fabian, D.: Alexandria: Unsupervised high-precision knowledge base construction using a probabilistic program. In: *AKBC* (2018)