Multimed Tools Appl
DOI 10.1007/s11042-012-1094-0

Variable linkage for multimedia metadata schema matching

Jordi Nin · Ruben Tous · Jaime Delgado

Abstract Today there are many media sharing applications that use diverse metadata formats to describe media resources. This leads to interoperability issues in cataloguing, searching and annotation. This situation poses schema matching algorithms in the eye of the storm of metadata interoperability. In this paper we present two different solutions for multimedia metadata schema matching using variable linkage algorithms. These methods consist in directly comparing the data values stored in the different metadata variables, allowing to overcome the inherent limitations of schema-level matching approaches. We show the feasibility of these methods through some experiments with real metadata information extracted from the image hosting websites Deviantart, Flickr and Picasa.

Keywords Metadata integration · Image tagging · Variable integration · Schema matching · Record linkage

1 Introduction

The rapid growth of multimedia content has been accompanied by a proliferation of metadata formats. Even within the same subject domain or for the same type of resource (e.g. digital images), there are often two or more options of metadata standards. Only if mechanisms can be developed to attain interoperability it will

J. Nin (✉) · R. Tous · J. Delgado
Departament d'Arquitectura de Computadors,
Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: nin@ac.upc.edu

R. Tous
e-mail: rtous@ac.upc.edu

J. Delgado
e-mail: jaime.delgado@ac.upc.edu

Published online: 29 April 2012

be possible to facilitate the exchange and sharing of content annotated according to different metadata formats and to enable cross-collection searching. Automatic schema matching [16] is a fundamental task in multimedia metadata interoperability approaches [4, 20, 21].

Schema matching seeks to provide automated support to the process of finding correspondences between two or more data schemas. Generalizing the schema concept, beyond its traditional binding to relational databases, the schema matching problem also embraces the ontology alignment topic [6]. Schema matching relies on the combination of several strategies which are usually classified within schema-level matching [16] or record-level matching [1]. While schema-level matching only considers information about the structure (hierarchy) and syntax of the involved schemas elements, record-level matching takes profit from data records to infer the possible relationships between schema variables. In general, any schema matching process has three main steps:

- **Pre-integration**. An initial analysis of the schemas to be integrated. Usually, this step includes the necessary data cleaning and normalization processes.
- **Schemas Comparison**. it determines the correspondences among variables and detects possible links.
- **Schemas Merging**. Once links are detected, variables are merged inside an integrated schema.

Schema-level matching methods allows for fast comparisons among different schemas. However, they have two important drawbacks: on the one hand, they assume that metadata schemas are correctly employed by the users, for instance imagine we would like to upload an image into an online web image repository. Schema-level matching methods will assume that we fill up the image title in the right web form field, but we may use the title field to annotate some keywords to make the subsequent image retrieval processes easier. On the other hand, such methods suppose that common information in two schemas is labeled with a similar description, hierarchy or variable name. However this is not always the case, and sometimes, similar information is labeled in a very different way (e.g. ‘format’ vs ‘file_extension’). Another important issue to take into account is that words like ‘source’, ‘description’, etc. are widely used within schemas and their meaning could have sense in very different domains. Usually, schema-level approaches match together all the variables containing these polysemous words.

Also, there are algorithms which perform record-level matching on its own, focused on inferring correspondences between data records instead of schema elements. These algorithms fall within the record linkage topic [5], and they are also known as variable linkage algorithms. These methods establish relationships between data objects (records described in terms of variables) that while supplied by different information sources (schemas in our scenario) correspond to the same entity. If the per-record match results are merged and abstracted to the schema level, a variable linkage algorithm becomes a solution to the record-level schema matching problem. Usually, record-level methods use record distances or probabilities calculations to define correct matchings.

Schema matching methods can be also classified as unsupervised, semi-supervised or supervised depending on the level of human interaction they have. Usually, unsupervised methods are used as an aid for further human integration and they only cover the first two steps of a complete schema matching process, leaving the last step (schemas merging) for human or experts task. However, they have the advantage of being able to do a large part of the work without human interaction.

In this paper, we present two unsupervised record-level schema matching approaches for multimedia metadata schemas matching based on variable linkage. The first one uses aggregation functions to preprocess and represent records in a common domain. The second one employs distribution fitting methods to extract some structural information about the records content making possible the re-identification. Once the metadata information has been preprocessed, both methods use the classical distance-based schema matching approach to determine the links. In order to test the feasibility of these two methods, we present a set of experiments with real metadata information extracted from three web image repositories (Flickr, Picasa and Deviantart). Here, We would like to highlight that the variable links found by our techniques are bijective, i.e. the schema order does not affect to the linkages found. This property is possible because the distances we use in both methods to determine the links are symmetric. For completeness, we will also compare our two methods with the current state-of-the art schema-level matching methods.

The rest of this paper is organized as follows. In Section 2 we present the new approaches for metadata schema matching. We also provide the preliminaries and the related work needed to make the understanding of the proposals easier. Next, in Section 3, we provide an experimental evaluation of the methods with real image metadata information. The work is concluded in Section 4.

2 Variable-level schema matching scenarios

A schema A can be thought as a matrix with n rows (*records*) and v columns (*variables*). Each row contains the values of the variables for some records (images, video or any multimedia content in general). The goal of any schema matching technique is therefore to compare two schemas A, B and find pairs (one in A, one in B) which correspond to the same variable.

Firstly, we introduce the distance-based approach. This widely used method uses distances to find common links. Distance-based approach is the fundamental building block of our two proposals: aggregation-based approach, where distances are calculated after records have been processed using a set of aggregation functions; and distribution-based approach, where finding the linkages is possible by means of variable distribution comparisons.

2.1 Distance-based approach

One possible option for metadata integration is distance-based (DB) variable linkage: for each variable v in one of the schemas, one searches the variable(s) in the others schemas that is (are) at minimum distance to v, for some distance defined on the domain of the variables. For the case of only two schemas, this operation is formalized as: for all $a \in A$, the subset $B_a = \{b' \in B \text{ s.t. } d(a, b') \leq d(a, b), \forall b \in B\}$ is found. Here a and b stand for a variable of the schema A and B respectively and $d(a, b)$ is a distance between a variable of the database A and a variable of the database *B*. The general (more than two schemas) pseudo-code of this algorithm is given in Algorithm 1.

Algorithm 1: DB-variable linkage

```
Data: $S_i$: schemas ($i \in \{1, \ldots, n\}$)
Result: LV: linked variables
begin
    for $i \leftarrow 1$ to $n$ do
        foreach $a \in S_i$ do /* $a$ stands for a variable of the schema $S_i$ */
            $b' = arg\_min_{b \in S_{j \neq i}} d(a, b)$ /* $j \in 1, \ldots, n$ */;
            $LV = LV \cup (a, b')$;
```

Obviously, the application of this algorithm is only possible if such distance function can be calculated. Normally, this distance depends on the type of variables to be linked. For instance, for numerical variables the Euclidean or Mahalanobis [10] distances are commonly used. For nominal variables, traditional distances as the Edit [3, 9] or Jaro [8] distances are the most used, however other distances as [7] can be also used. For categorical and ordinal variables (i.e. academic degree, zip code, etc) specific distances seem the best choice because their values are strongly related with their semantic. For example, for zip code distance the geodesic distance between two zip codes or their hamming distance (amount of different numbers) can be good candidates. Also, distances has only sense when they are computed over the same set of records.

In Algorithm 1 we have implemented a top-k (with $k = 1$) query, i.e. we have assumed as correct linkages all the variable pairs among schemas at the minimum distance. This configuration seems correct in our scenario where metadata formats share a large number of common variables. An alternative method for deciding which variables have to be linked is to accept all the linkages with a distance below a certain threshold t. In this case an expert should fix this threshold because the quality of the results strongly depends on it.

The complexity of this method is equal to $O(s \cdot v^2)$ where s stands for the number of schemas and v for the number of variables of the biggest schema. As v is usually not very big (<100) the performance of this algorithm relies on the distance cost. For instance, edit distance has a cost of $O(n^2)$ where n is the length of the largest variable to compare. Then the real cost in this case will be $O(n^2 \cdot s \cdot v^2)$.

Note that, this approach has a very important drawback: it is necessary that schemas share some common records, images in our target scenario. Moreover, we have to know in advance which concrete records are and how they are related. Distances have only sense when they are computed over the same data elements, for this reason DB-variable algorithm cannot be applied in scenarios where this knowledge is not available. The two following improvements described in this section overcome this problem.

2.2 Aggregation-based approach

In scenarios where it is not possible to have some common records inside the schemas to be integrated or we do not know how records are related, different approaches should be considered. In [17] OWA operators [23] were used to find a variable data model from the records stored in several databases. Later, in [19] a more general procedure for record linkage was presented. In this latter work, authors show that record linkage is still possible even when databases do not share common variables. The same reasoning can be extended to multimedia metadata integration as we will see later.

Then, a possibility to overcome the problem of the distance-based approach is to adapt the proposal in [19] to the metadata format matching scenario. Before that, we need to introduce some basic concepts about aggregation functions (Section 2.2.1). After that, in Section 2.2.2 the aggregation-based approach is completely described.

2.2.1 Aggregation functions basics

Aggregation functions [18] (Definition 1) are numerical functions used for information fusion that combine n numerical values into a single one. These functions, that are formally described below, typically satisfy unanimity (idempotency) and monotonicity.

Definition 1 Let $X := \{x_1, \dots, x_n\}$ be a set of information sources, and let $f(x_i)$ be a function to model that the i-th information source x_i supplies value $f(x_i)$, then a function $\mathbb{C} : \mathbb{R}^n \to \mathbb{R}$ to aggregate values $f(x_i)$ is said to be an aggregation function if it satisfies:

1. $\mathbb{C}(a, \dots, a) = a$ (unanimity, also known as idempotency)
2. $\mathbb{C}(a_1, \dots, a_n) \leq \mathbb{C}(a'_1, \dots, a'_n)$ if $a_i < a'_i$ (monotonicity)

At present, several aggregation functions exist in the literature (see [18] for a review). Among them, the most well-known aggregation functions are the arithmetic mean and the weighted mean. They correspond, respectively, to the following functions:

1. $\mathbb{C}(a_1, \dots, a_n) = \frac{\sum_i^n a_i}{n}$
2. $\mathbb{C}(a_1, \dots, a_n) = \sum_i^n w_i a_i$

In the second definition, $\mathbf{w} = (w_1 \dots w_n)$ stands for a weighting vector. That is, w_i are weights for sources x_i such that $w_i \geq 0$ and $\sum_i w_i = 1$. These values correspond to prior knowledge on the reliability of the sources. For example, when source x_i is twice as reliable as source x_j then we have that $w_i = 2w_j$.

Yager [23] defines the so-called Ordered Weighted Averaging (OWA) operator (Definition 2) that corresponds to a weighted linear combination of order statistics. At present there are different definitions for this operator based on the way the weights are defined. In this paper, we recall a definition based on a non-decreasing function, as this is the most useful definition in our context.

Definition 2 Let Q be a non-decreasing function in $[0, 1]$ such that $Q(0) = 0$ and $Q(1) = 1$, then the mapping $OWA_Q : \mathbb{R}^n \to \mathbb{R}$ defined as follows is an OWA operator:

$$OWA_Q(a_1, \dots, a_n) = \sum_{i=1}^{n} \big(Q(i/n) - Q((i-1)/n)\big) a_{\sigma(i)}$$

where σ is a permutation of the values a_i such that $a_{\sigma(i)} \geq a_{\sigma(i+1)}$.

This function has several properties. We underline the following ones:

1. For all Q, it holds that:

$$\min_i a_i \leq OWA_Q(a_1, \ldots, a_n) \leq \max_i a_i.$$

2. The function Q permits to modulate the output. For example, when we consider the family of functions $Q_\alpha(x) = x^\alpha$, we have that large positive values of α lead to an OWA near to the minimum and, instead, values of α near to zero lead to an OWA near to the maximum. Also, when a_i is fixed, OWA_{Q_α} is non-decreasing with respect to α.
3. The OWA operator is symmetric for all Q. That is, the order of the parameters is not relevant for the computation of the output. This can be formalized as follows:

$$OWA_Q(a_1, \ldots, a_n) = OWA_Q(a_{\pi(1)}, \ldots, a_{\pi(n)})$$

for any permutation π.

2.2.2 Integration procedure

In order to apply aggregation functions to the metadata interoperability scenario we have to consider the following assumptions:

Assumption 1 A set of common variables is shared by both metadata formats.

If this assumption does not hold, then variable linkage has no sense because the metadata formats are completely different.

Assumption 2 Data in both schemas contains, implicitly, similar structural information.

In metadata interoperability, we can define 'metadata structural information' as any organization of the data that allows explicit representation of the existing relationships among variables. Such information can be extracted from schemas through record-data manipulation. In other words, even though we do not know any concrete link between the common records, there is substantial correlation among them to establish some relations.

As different methods can be used for representing such structural information (e.g. clustering, principal component analysis, or any other data mining method), different techniques are needed to extract structural information. Here, we focus on structural information represented by means of numerical representatives. Therefore, we have to add the following assumption to the previous ones:

Assumption 3 Structural information is expressed by means of numerical representatives for each record.

A representative is the output of a concrete aggregation function once its weighting vector and input, in our case the record values of a variable, have been set up. Torra and Nin [19] shows that only few aggregation functions are appropriate for building such representatives. One of them is the OWA operator described before, however for the sake of generality in our proposal we will consider any function $f \in \mathcal{F}$ as valid.

To tackle the problem of metadata integration, we consider the transformation of schemas A and B into two new schemas A' and B' with the goal that DB-variable linkage algorithm can be applied on this second pair of schemas (A', B').

To do so, we consider the construction of several representatives for each variable a in A and each variable b in B so that integration can be performed over such representatives. This process is detailed below:

- Firstly, we consider a set of functions f_i for building the representatives. In general, we consider that f_i is a function of both the variable and of the whole schema A. Therefore, being a a variable in A, $f_i(a, A)$ stands for a representative of the variable a of the schema A. We denote by $\mathcal{F} = \{f_i\}$ for $i = 1, \ldots, k$ the set of considered functions.
- Then, we apply the functions in $\mathcal{F}$ to the variables a in A to obtain a'. Formally speaking $a' := \mathcal{F}(a, A)$ where:

$$a' := \mathcal{F}(a, A) = (f_1(a, A), \ldots, f_k(a, A))$$

- Now, assuming that functions in $\mathcal{F}$ are also applicable to variables b in B, we define variables b' in B in a similar way:

$$b' := \mathcal{F}(b, B) = (f_1(b, B), \ldots, f_k(b, B))$$

- Finally, we define schemas A' and B' in terms of the new variables a' and b'. That is:

$$A' := \{\mathcal{F}(a, A)\}_{a \in A}$$

$$B' := \{\mathcal{F}(b, B)\}_{b \in B}$$

Therefore, given the set of functions $\mathcal{F} = \{f_i\}$ for $i = 1, \ldots, k$, and applying each f_i to each variable in A and B, we obtain schemas A' and B'. This process is defined in the Algorithm 2. The number of representatives, i.e. the number of aggregation functions and weighting vectors to consider, is a parameter that has to be decided in advance. In our experiments after some tests we have found that the obtained results improve until 20 representatives, from here the results are constant. Therefore, 20 representatives seems a good configuration for our target scenario.

Algorithm 2: transformSchema

```
Data: A: schema, $\mathcal{F}$: set of functions
Result: A': transformed schema
begin
    foreach $a \in A$ do
        a'=new_record($f_1$(a,A), ..., $f_k$(a,A));
        write(a',A');
end
```

With this construction, A' and B' contain the same number of variables as A and B, and such variables in both schemas are described using the same kind of representatives (records). Now, as we know the links between the representatives DB-variable linkage algorithm can be applied to the pair (A', B').

2.3 Distribution-based approach

A different approach when it is not possible to know some records links is to compare variable distributions. Such comparison can be done in several ways. For example, it is easy to see that for categorical variables, categories can be used directly to fit the variable distribution. Similarly for nominal variables, the words or characters frequency can be used as the categories in the case of categorical variables.

Now, we introduce some basic concepts about variable distribution fitting before describing the distribution-based approach for metadata formats integration.

2.3.1 Distribution fitting

In statistical literature a data distribution describes the frequency or probability of possible events, where an event is a set of possible outcomes of an experiment, or in other words, it is a sequence of observations. Such observations are considered as the data. The process of finding the correct data distribution that describe the frequencies of future or non-observed outcomes is very useful in a large variety of domains like missing value imputation [14], time series forecast [12], etc.

Fitting distributions consists in finding a mathematical function which represents in a good way a variable. We can identify four steps in fitting distributions:

1. **Model/function choice**: exploratory data analysis can be the first step, getting descriptive statistics (mean, standard deviation, skewness, etc.) and using graphical techniques (histograms, density estimate, etc.) which can suggest the kind of probability distribution function (PDF) to use to fit the variable model.
2. **Estimate parameters**: After choosing a model that can mathematically represent the data, we have to estimate the parameters of such model. There are several estimate methods in statistical literature, as for instance maximum likelihood.
3. **Evaluate quality of fit**: A goodness of fit measure is useful for matching empirical frequencies with fitted ones by a theoretical model. In this case, we have several measures, as for instance the Sum of Square Errors (SSE).
4. **Goodness of fit statistical tests**: Goodness of fit tests indicates whether or not it is reasonable to assume that a variable comes from a specific distribution. The chi-square test or some normality tests are widely used.

Now, we introduce the maximum likelihood method for distribution parameter estimation: we have a random variable with a known $PDF f(a, \Theta)$ describing a quantitative parameter in a database. We should estimate the vector of constant and unknown parameters Θ according to observed data: $a_1, a_2, ...a_n$. Maximum likelihood estimation begins with the mathematical expression known as a likelihood function of the sample data. Roughly speaking, the likelihood of a set of data is the probability of obtaining that a particular set of data given the chosen probability model. This expression contains the unknown parameters. Those parameter values maximizing the sample likelihood are known as the maximum likelihood estimates (MLE). We define the likelihood function as:

$$L(a_1, a_2, \ldots, a_n, \Theta) = \Pi_{i=1}^{n} f(a, \Theta)$$

MLE consists in finding Θ which maximizes $L(a_1, a_2, \ldots, a_n, \Theta)$ or its logarithmic function. We can employ mathematical analysis methods (as partial derivates equal to zero) when the likelihood function is rather simple, but very often we optimize $L(a_1, a_2, \ldots, a_n, \Theta)$ using iterative methods like the gradient descent method.

2.3.2 Integration procedure

In order to use distribution fitting techniques to integrate two metadata formats we have to proceed as follows:

- Firstly, for all the variables a, b of the schemas A, B respectively we extract their mean (μ) and variance (σ^2), as well as, we plot its corresponding histogram.
- Then, we apply the aforementioned method of maximum likelihood to estimate the distribution parameters of all variables of the schemas. As the obtained distributions maximize the expression $L(a_1, a_2, \ldots, a_n, \Theta)$ or $L(b_1, b_2, \ldots, b_n, \Theta)$ for the schema variables a and b respectively, we assume that those distributions are the best fitted ones.
- Finally, we have to compare the obtained distributions in a graphical or analytical way to decide the variable matches.

As the amount of schema variables used in the experiments carried out in Section 3 are quite small, we have done this last step of distribution comparison manually using a top-1 query format as in the case of DB-variable linkage. However, any more complex decision making algorithm based on multi-criteria decision analysis (MCDA) and its varieties [2] can be used to solve this step of the distribution-based approach when the amount of variables increases.

3 Experiments

In this section we describe the experiments we have carried out. Firstly, we describe the metadata extraction process we performed, then we introduce the performance measures we have applied. After that, we discuss the obtained results using the aggregation-based and distribution-based approaches.

3.1 Metadata extraction process

In our experiments we have used real metadata formats and records from different web images repositories, such as Deviantart (http://www.deviantart.com/), Flickr (http://www.flickr.com/) or Picasa (http://picasa.google.com/). Such websites provide public APIs which allow us to retrieve different sets of public image metadata to work with. Table 1 summarizes the selected variables, the provided names are represented using XPath notation. As we have extracted the metadata records executing a different XML queries in each server through their APIs, it was impossible to automatically determine the links among the different images retrieved. Indeed, it is even possible that those repositories do not share any common image. Therefore we cannot apply the distance-based approach. We would like to highlight that the metadata formats we used in this work are flat schemas. Hierarchical/XML schemas are not covered in this experiments.

Table 1 Summary of the variables selected of Deviantart, Flickr and Picassa metadata schemas

Aggregation	Deviantart	Flickr	Picasa
Author/name (text)		Photo/owner@realname	Entry/author/name
Author/nick (text)	Item/media:credit [@role='author']	Photo/owner@username	Entry/author/uri
Image/height (numeric)	Item/media:content [@medium='image'] @height	Sizes/size[@label= 'Medium']@height	Entry/gphoto/height
Image/width (numeric)	Item/media:content [@medium='image'] @width	Sizes/size[@label= 'Medium']@width	Entry/gphoto/width
Image/exif/make (text)		Photo/exif[@label= 'Manufacturer']/raw	Entry/exif:tags/exif:make
Image/exif/model (text)		Photo/exif[@label= 'Model']/raw	Entry/exif:tags/exif:model
Title (text)	Item/title	Photo/title	Entry/title

Aggregation stands for the given name of each variable in the aggregated metadata format

For the sake of fairness we have queried the databases using several keywords. Concretely, we have executed five different queries. The first one completely random (without keywords) and the other four using two general keywords (weekend and trip) and two particular (Barcelona and London). Doing this we want to ensure that our results do no depend on the executed query. For each query we have retrieved 1,000 completely tagged images.

For completeness of the experiments we have mixed nominal and numerical variables in the same schema. Surely, if we separate the variables by type, one schema for nominal variables and another for numerical ones, the obtained results would be better, however we want to test our approaches in a more complex and real scenario.

In order to apply the aggregation-based approach and to fit the variable distributions of the distribution-based approach, for the nominal variables we have considered the numerical ASCII code of each letter. Of course, more complex conversions are also possible, however data normalization / standardization is out of the scope of this work and for this reason in this point we have chosen the easiest solution.

3.2 Performance measures

After describing the metadata used in the experiments, we describe the quality measures we have used. As schema matching problems are specific problems of machine learning [11], we have used the common measures of classification tasks [22]. Usually, the terms *true positives*, *true negatives*, *false positives* and *false negatives* compare the predicted class of an item (the class label assigned by a classifier) with the actual class. This is illustrated by the Table 2. By means of these terms we can formulate the following measures:

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{1}$$

Table 2 Classification results

	Real class	
Predicted class	tp (True positive)	fp (False positive)
	fn (False negative)	tn (True negative)

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{2}$$

Precision is the fraction of items correctly classified. Recall, also called true positive rate, represents the fraction of correctly classified items that are successfully classified.

Sometimes it is desirable to have one single number for the performance of an algorithm instead of two. In such cases, the F-measure is frequently used [13]. It can be parameterized to give a higher weight to either precision or recall. The neutral parameterization, where precision and recall are weighted equally, is used throughout this work. Thus, F is defined as the weighted harmonic mean of precision and recall:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

It is important to note that in our experiments we have only considered as true positive the variables correctly linked in all the web image repositories at the same time. For instance, the variable author/nick is only considered as correctly classified when it is correctly linked in the three repositories (Deviantart, Flickr and Picasa). On the other hand, the image/exif/make variable is considered as correctly classified if it is correctly linked only between Flickr and Picasa as it does not appear in the Deviantart metadata format.

3.3 Results analysis

Now, we analyze the obtained results, to do this we have divided the analysis into two parts, one for each of the approaches we have tested.

3.3.1 Aggregation-based approach

Experiments with the aggregation-based approach have been done using the OWA operator described in Section 2.2.1. Two completely different families of non-decreasing functions were considered for building the weighting vectors. The functions and the parameters used are the following ones:

1. $Q^e_\alpha(x) = x^\alpha$ for $\alpha = 1/5, 2/5, 3/5, \ldots, 10/5$
2. $Q^s_\alpha(x) = 1/(1 + e^{(\alpha - x)*10})$ for $\alpha = \{0, 0.1, \ldots 0.9\}$

Here, Q^e stands for exponential function and Q^s for sigmoidal function. Therefore, we have created 20 different representatives, ten for each function family. To do so, in this scenario we have normalized the ASCII values into the [0, 1] interval.

After that we have applied on these representatives the DB-variable linkage algorithm (Algorithm 1) configured as a top-1 query, i.e. we have only considered the link at minimum distance as the correct one. This configuration has a lot of sense in our scenario because it is not usual to have variables divided into several parts inside a single metadata format.

Table 3 illustrates the results we have obtained. Due to the fact we are considering the top-1 query configuration the precision and recall values we obtained are equal. Despite of this, the obtained results show that query specificity does not affect too much the results. Indeed, the more random the query, the better the results.

We would like to highlight the quality of the results obtained. For instance, for the trip query, we have obtained a F-measure equal to 0.600, this means that we have correctly linked five of the eight attributes among the three schemas without any kind of human supervision.

Further considerations about the concrete obtained linkages are depicted in the following section.

3.3.2 Distribution-based approach

To study the performance of the distribution-based approach we have conducted the same experiment than in the aggregation-based approach. In this case, we have used the implementation of the maximum likelihood method available in R software (The R Project for Statistical Computing http://www.r-project.org/) for the distribution parameters estimation. We have assumed that most of the variables follow a normal distribution after observing their histograms in detail. As an example of the obtained results, in Table 4 we show the obtained parameters of the variable distribution assuming that variables follow a normal distribution for the query related with Barcelona images.

After that to automatically link the variable distributions we have normalized the obtained μ and σ^2 parameters into the interval [0, 1] and again we have used the DB-variable linkage with a top-1 configuration to establish the variable links. As it is depicted in Table 5, the obtained results are exactly the same for all the queries. This fact shows that the distribution-based approach is in some sense independent of the performed query.

Comparing the F-measure values obtained by both methods we can say that distribution-based approach works better than the aggregation-based approach. In our opinion this happens because in the schemas there are more nominal variables than numerical (five nominal vs. only two numerical). This has sense because the numerical ASCII code conversion and normalization can affect to the strong character distribution correlation among the variables.

Using distribution-based approach, it is possible to link all the variables except the title variable. These very good results are possible due to the very strong relation

Table 3 Performance classification results for the aggregation-based approach

Query	Precision	Recall	F-measure
Random	0.567	0.567	0.567
Weekend	0.533	0.533	0.533
Trip	0.600	0.600	0.600
London	0.433	0.433	0.433
Barcelona	0.433	0.433	0.433

Table 4 Variable distribution parameter values

	Variable	Type	μ	σ
Deviantart	Author/nick	Text	92.44	25.15
	Image/height	Numeric	680.94	240.50
	Image/width	Numeric	752.07	244.13
	Title	Text	92.48	25.12
Flickr	Author/name	Text	95.82	25.11
	Author/nick	Text	100.66	33.81
	Image/height	Numeric	1195.92	758.615
	Image/exif/make	Text	81.47	19.18
	Image/exif/model	Text	70.44	23.41
	Image/width	Numeric	1409.10	906.66
	Title	Text	92.48	25.12
Picasa	Author/name	Text	98.64	21.16
	Author/nick	Text	101.81	17.58
	Image/height	Numeric	1108.31	463.42
	Image/exif/make	Text	82.42	19.23
	Image/exif/model	Text	71.83	23.67
	Image/width	Numeric	1342.98	588.34
	Title	Text	72.01	24.96

among common variables in all the metadata formats. From the obtained links (Table 5) and the results depicted in Table 4 it is possible to infer the following conclusions:

– The distributions of the metadata variables author/name, image/exif/make and image/exif/model are almost identical in the three databases, then they are very easy to link. This fact has lot of sense because there are a limited number of camera manufactures and models. A similar reasoning also applies for the user's name field (author/name).
– The distributions of the author/nick variable have the same μ in the three repositories, however as Flickr allows the use of special characters its standard deviation is larger than in the other two websites. Picasa also allows for special characters but as the author/nick variable is placed in the uri of the image, then they are escaped in the retrieving process. For that, linkage is more difficult but still possible.
– A very interesting issue occurs with the title variable distribution. The corresponding Picasa variable follows a very different statistical distribution from Flickr and Deviantart. On the light of this result one could think that something

Table 5 Performance classification results for the distribution-based approach

Query	Precision	Recall	F-measure
Random	0.857	0.857	0.857
Weekend	0.857	0.857	0.857
Trip	0.857	0.857	0.857
London	0.857	0.857	0.857
Barcelona	0.857	0.857	0.857

is wrong in the approach or in the implementation. However, after checking the content of the Picasa metadata schema, one discovers that Picasa website is using this metadata variable for storing the file name instead of the image title. Therefore, it is correct that variables do not follow the same distribution.

From the last observation, a new possible application for our algorithms arises: it is possible to use them as a consistency rule for ensuring a correct use of the metadata model. This kind of methods are very useful for transactions rules registers, query rewriting tools, traditional schema matching algorithms (based on variable names and/or ontologies), etc.

3.4 Comparison with different approaches

In order to compare the results obtained by our proposals with the ones that could be obtained by state-of-the-art methods, we have executed two sets of experiments, one with record-level methods and another schema-level methods. In the first set of experiments, we have executed the basic distance-based approach (DB-variable algorithm, Section 2.1) and the probabilistic-record linkage (PRL) approach described in [8] over the same five queries. The results of these executions are depicted in Table 6. As we observe, classical record-level methods do not perform as well as the ones proposed in this work in our target scenario. Classical methods only are able to link one or two variables at most in all the considered queries.

For the second set of experiments, we have used the schema-level toolbox called OpenII [15]. This toolbox is a complete suite of open-source methods for schema integration; and it implements a large variety of schema-level matchers based on variable name similarity functions, ontology alignment methods (e.g wordnet matcher), hierarchical variable relations, etc.

For the sake of fairness, we have executed all the variable matchers implemented in OpenII over the Deviantart, Flickr and Picasa metadata schemas. Note that, as OpenII matchers only takes into account schema metadata information, the retrieved records content for the queries is discarded and the F-measure results are exactly the same for all the queries. From the links obtained by OpenII matchers, we have calculated by hand the F-measure using the best matcher in the same way than in our previous experiments. Concretely the obtained F-measure is equal to 0.571, i.e. openII is able to correctly link four of the seven schema variables. This F-measure value is quite good, indeed, it is in the same order of magnitude than F-measure results obtained by the aggregation-based approach. However, when these values are compared with the distribution-based approach values (F-measure equal to 0.857), the distribution-based approach clearly outperforms openII results.

Table 6 F-measure values obtained by traditional record-level methods

Query	DB-variable algorithm	PRL algorithm
Random	0.28	0.28
Weekend	0.14	0.28
Trip	0.14	0.28
London	0.14	0.14
Barcelona	0.28	0.28

4 Conclusions

In this work we have analyzed two variable linkage algorithms as possible strategies for performing record-level schema matching in the context of the digital images metadata multimedia interoperability problem. Apart from describing the algorithms, we have provided the necessary mathematical background and a detailed related work for those methods. Also, we have performed some interoperability experiments with three real well-known image websites (Deviantart, Flickr and Picasa). In these experiments, we have illustrated how to apply the aggregation-based and distribution-based schema matching approaches to real data. We have obtained very good results using the distribution-based approach. Also, with the experiments done, we have also discovered that Picasa website uses the field title to store the image file name instead of the real title image.

As future work we would like to propose the methods described here as a possible solution for the automation of some of the administrative processes derived from the translation rules registers.

Acknowledgements This work has been partly supported by the Spanish government TEC2008-06692-C02-01 and ARES CONSOLIDER INGENIO 2010 CSD2007-00004.

References

1. Berlin J, Motro A (2002) Database schema matching using machine learning with feature selection. In: 14th int. Conf. of Advanced Information Systems Engineering (CAiSE). Lecture notes in computer science, vol 2348, pp 452–466
2. Bouyssou D, Marchant T, Pirlot M, Tsoukias A, Vincke P (2011) Evaluation and decision models with multiple criteria: stepping stones for the analyst. In: International series in operations research & management science, vol 86. Springer
3. Damerau FJ (1964) A technique for computer detection and correction of spelling errors. Commun ACM 7(3):171–176
4. Doeller M, Stegmaier F, Kosch H, Tous R, Delgado J (2010) Standardized interoperable image retrieval. In: Proceedings of the 2010 ACM symposium on applied computing. SAC '10. ACM, New York, pp 880–886
5. Elmagarmid AK, Ipeirotis PG, Verykios VS (2007) Duplicate record detection: a survey. IEEE Trans Knowl Data Eng (TKDE) 19(1):1–16
6. Euzenat J, Shvaiko P (2007) Ontology matching. Database management & information retrieval. Springer
7. Herranz J, Nin J, Solé M (2011) Optimal symbol alignment distance: a new distance for sequences of symbols. IEEE Trans Knowl Data Eng (TKDE) 23(10):1541–1554
8. Jaro M (1989) Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. J Am Stat Assoc 84:414–420
9. Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl 10:707–710. http://www.freearchive.org/o/964e741877a3ffa8f2195ee253597fbaeed69a207e77df150439802fd370b2f8
10. Mahalanobis P (1936) On the generalised distance in statistics. In: Proceedings of the national institute of sciences of India, vol 2, pp 49–55
11. Mitchell T (1997) Machine learning. McGraw-Hill
12. Nin J, Torra V (2009) Towards the evaluation of time series protection methods. Inf Sci 179(11):1663–1677
13. Rijsbergen C (1979) Information retrieval. Butterworth
14. Rubin D (1976) Inference and missing data. Biometrika 63:581–590
15. Seligman L, Mork P, Halevy A, Smith K, Carey M, Chen K, Wolf C, Madhavan J, Kannan A, Burdick D (2010) Openii: an open source information integration toolkit. In: ACM int. conf. on management of data (SIGMOD), pp 1057–1059

16. Shvaiko P, Euzenat J (2005) A survey of schema-based matching approaches. J Data Semantics IV, LNCS 3730:146–171
17. Torra V (2004) OWA operators in data modeling and reidentification. IEEE Trans Fuzzy Syst 12(5):652–660
18. Torra V, Narukawa Y (2007) Modeling decisions: information fusion and aggregation operators. Springer
19. Torra V, Nin J (2008) Record linkage for database integration using fuzzy integrals. Int J Intell Syst (IJIS) 23(6):715–734
20. Tous R, Delgado J (2009) A lego-like metadata architecture for image search & retrieval. In: Proceedings of the 2009 20th international workshop on database and expert systems application, pp 246–250
21. Tous R, Nin J, Delgado J (2011) Approaches and standards for metadata interoperability in distributed image search&retrieval. In: 22nd Int. conf. on database and expert systems applications. Lecture notes in computer science, vol 6861. Springer, pp 234–248
22. Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques. Morgan Kaufmann
23. Yager R (1988) On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans Syst Man Cybern 18:183–190

Jordi Nin holds a Ph.D. (2008) in Computer Science by the Autonomous University of Barcelona (UAB). He works as a tenure-track Lecturer in the Computer Architecture Department of the Universitat Politècnica de Catalunya (UPC). Previously, from 2009 to 2010 he was a Marie Curie postdoc reseacher in the Laboratoire d'Analyse et d'Architecture des Systemes (LAAS) in Toulouse, France. From 2007 to 2009 he worked as pre-doctoral researcher at the Artificial Intelligence Research Institute (IIIA) of the Spanish National Research Council (CSIC). His fields of interests are privacy technologies, machine learning and soft computing tools. He has been involved in several research projects funded by the Catalan and Spanish governments and the European Community. His research has been published in specialized journals and major conferences (around 50 papers).

Ruben Tous received his Ph.D in Computer Science and Digital Communication from UPF (Universitat Pompeu Fabra, Barcelona, Spain) in 2006. From 2000 to 2001 he worked as a consultant at CapGemini Ernst&Young in Barcelona, From 2001 to 2005 he worked at the Department of Techology of UPF. Since 2006 he is a researcher at DMAG (Distributed Multimedia Applications Group) of the Department of Computer Architecture of UPC (Universitat Politecnica de Catalunya, Barcelona, Spain) and an Associate Professor. He is an expert for the AsociaciÛn EspaÒola de NormalizaciÛn y CertificaciÛn (AENOR) and has been participating as spanish delegate in ISO/MPEG and ISO/JPEG. His research interests include algorithms and data structures, knowledge representation and reasoning for multimedia understanding, multimedia databases and query languages, multimedia information retrieval and image retrieval in medical applications.

Jaime Delgado received his Ph.D. degree in Telecommunication Engineering in 1987. Since September 2006, Professor at the Computer Architecture Department of the Universitat Politècnica de Catalunya (UPC) in Barcelona (Spain). Previously, Professor at the Universitat Pompeu Fabra (UPF), since 1999. Head and founder of the Distributed Multimedia Applications Group (DMAG). Project Manager of several European and national research projects. Active participation, since 1989, in International standardisation, as editor of standards and chairman of groups. Evaluator and reviewer for the European Commission, since 1989, and several Spanish Ministries. Author of a few hundreds of published papers and books.