The VLDB Journal (2008) 17:1371–1384
DOI 10.1007/s00778-008-0103-4

SPECIAL ISSUE PAPER

PicShark: mitigating metadata scarcity through large-scale P2P collaboration

Philippe Cudré-Mauroux · Adriana Budura · Manfred Hauswirth · Karl Aberer

Received: 23 September 2007 / Revised: 27 February 2008 / Accepted: 9 May 2008 / Published online: 15 July 2008

Abstract With the commoditization of digital devices, personal information and media sharing is becoming a key application on the pervasive Web. In such a context, data annotation rather than data production is the main bottleneck. Metadata scarcity represents a major obstacle preventing efficient information processing in large and heterogeneous communities. However, social communities also open the door to new possibilities for addressing local metadata scarcity by taking advantage of global collections of resources. We propose to tackle the lack of metadata in large-scale distributed systems through a collaborative process leveraging on both content and metadata. We develop a community-based and self-organizing system called PicShark in which information entropy—in terms of missing metadata—is gradually alleviated through decentralized instance and schema matching. Our approach focuses on semi-structured metadata and confines computationally expensive operations to the edge of the network, while keeping distributed operations as simple as possible to ensure scalability. PicShark builds on structured Peer-to-Peer networks for distributed look-up operations, but extends the application of self-organization principles to the propagation of metadata and the creation of schema mappings. We demonstrate the practical applicability of our method in an image sharing scenario and provide experimental evidences illustrating the validity of our approach.

Keywords Metadata scarcity · Metadata heterogeneity · Metadata entropy · Peer-to-Peer collaboration · Peer data management

1 Introduction

Until recently, the creation of digital artifacts—such as electronic documents, images, or videos—was constrained by the limited availability of devices capable of capturing and handling information in binary form. Today, the situation has radically changed with the commoditization of digital devices. Typewriters have now totally disappeared from the office space, whereas email has become one of the main communication channels. Mobile phones can handle information written as bidimensional bar-codes, while personal computers casually store and process gigabytes of personal images. In this new context, we argue that the lack of metadata, rather than the lack of data, has become the main bottleneck.

The problem became apparent a few years ago when end-users suddenly had to resort to third-party tools to find relevant pieces of information on their own computer. At that time, several projects proposed to index information based on metadata to enhance the search process. Microsoft's *Stuff I've Seen* [14], for instance, relies on time-stamp metadata like *Last Time Modified* or *Last Time Opened* to display search results, while Google Desktop[1] indexes documents based

The work presented in this article was supported by the Swiss NSF National Competence Center in Research on Mobile Information and Communication Systems (NCCR MICS, grant number 5005-67322), by the EPFL Center for Global Computing as part of the European project NEPOMUK No FP6-027705, and by the Líon project supported by Science Foundation Ireland under Grant No. SFI/02/CE1/I131.

P. Cudré-Mauroux (✉) · A. Budura · K. Aberer
School of Computer and Communication Sciences EPFL,
1010 Lausanne, Switzerland
e-mail: Philippe.Cudre-mauroux@epfl.ch

M. Hauswirth
Digital Enterprise Research Institute,
National University of Ireland, Galway, Ireland
e-mail: manfred.hauswirth@deri.org

[1] http://desktop.google.com/.

on metadata extracted from the files payload. Local search based on indexed metadata is considered as a common feature nowadays and has been integrated in most operating systems.

The lack of metadata resurfaces today as a new problem in distributed settings. More and more platforms allow end-users to share their digital content in large communities: Flickr, YouTube, and MySpace are well-known examples of that trend. In distributed environments, however, automatically generated metadata such as *Last Time Opened*, *Filename*, or *Size* often cannot be exploited in a meaningful way by arbitrary users searching for a specific file. In a distributed setting, users typically have never encountered the file they are searching for and are thus unaware of its technical details. Higher-level, more meaningful metadata like *Description*, *Event*, or *Location* are much more relevant in distributed environments, but still often require human attention, one of the scarcest resources in our digital society. As a result, the majority of digital content on current collaborative platforms simply cannot be retrieved by third-parties because of the lack of adequate metadata.

In the following, we tackle the problem of metadata scarcity in large-scale collaborative environments. We focus on semi-structured metadata formats and propose a radically new approach to foster global search capabilities from incomplete, local, and heterogeneous metadata. Our approach is based on a new metric for metadata scarcity and on Peer-to-Peer (P2P) interactions. The main contributions of our work are:

- the formalization of the problem of sharing semi-structured metadata in distributed settings, explicitly taking into account metadata incompleteness and metadata heterogeneity
- the definition of a new metric—called *metadata entropy*—to capture the degree of incompleteness or uncertainty related to semi-structured metadata
- the description of a bottom-up and recursive process based on instance and schema matching to infer metadata in collaborative P2P contexts
- the presentation of a system architecture supporting metadata inference in distributed environments
- the experimental evaluation of our metadata inference process on a large set of several hundreds of annotated images.

We start with a general description of the problems related to the sharing of semi-structured metadata in Sect. 2 and formalize our problem in Sect. 3. Our metadata sharing approach is presented in detail in Sect. 4. We describe the architecture of our prototype and the results of our experimental evaluation in Sect. 5. Finally, we give a survey of related work in Sect. 6 before presenting our conclusions.

2 Sharing semi-structured metadata

2.1 On semi-structured metadata

While the use of unstructured metadata drew considerable attention on the Web in recent years—e.g., through keyword annotation of images or HTML pages—the focus recently shifted back to more structured metadata formats. Unstructured metadata such as tags are ambiguous by nature and lack precise semantics, making it very difficult to support structured searches *à la* SQL. Structured representations such as relational tables are much easier to process automatically, as they constrain the representation of data through complex data structures and schemas.

In the following, we focus on recent formats that let end-users freely define and extend their own schemas according to their needs. We qualify those formats as *semi-structured* formats since they tend to blur the separation between the data and schemas and to impose looser constraints than the relational model to the data. Such formats are today sprouting from various contexts and encompass a large variety of data models. The Extensible Markup Language (XML) [6], for example, relies on hierarchies of elements to organize data or metadata. Ontological metadata tie metadata to formal descriptions where classes of resources (and properties) are defined and interrelated. This class of metadata standards is currently drawing a lot of attention with the advent of the Semantic Web and its associated languages (e.g., RDF/S [24], Adobe's XMP[2] or OWL [25]). Semi-structured formats are gaining momentum. They are flexible enough to allow easy definition and extension of schemas, while sufficiently structured to support automated processing and complex searches (e.g., through languages such as XQuery [5] or SPARQL [27]).

2.2 On the difficulty of sharing semi-structured metadata

Our goal is to enable global search capabilities for shared resources based on semi-structured metadata in large-scale, heterogeneous and distributed settings. Although semi-structured metadata formats are getting increasingly popular, support for meaningfully sharing semi-structured metadata outside of their original context or community of interest is often lacking. Semi-structured metadata are intrinsically difficult to share, since their values only make sense in a given context—as opposed to keyword metadata or textual tags, which supposedly convey predefined, global semantics. Hence, large-scale collaborative applications typically disregard semi-structured metadata or treat them as simple unstructured keywords ignoring their intrinsic structure.

[2] http://www.adobe.com/products/xmp/.

A straightforward solution to our problem would be to use a common language, like RDF, for all metadata. Though necessary, we argue that this syntactical alignment step only represents the tip of the iceberg in our case. Even with a global, common language, fundamental problems remain: systems would still be unable to retrieve all relevant resources given a query, as metadata can still be incomplete and unrelated one to another because of the various schemas introduced by the users.

In the end, two fundamental hurdles prevent semi-structured metadata from being shared meaningfully:

Metadata incompleteness: Though more and more tools rely on some semi-automatic annotation schemes to add metadata to resources, fully-automated solutions remain impractical. Most of the time, human attention is still required for producing high-quality, meaningful metadata. Realistically, a (potentially large) fraction of the shared resources will not be annotated by the user, leaving some (most) of the related semi-structured metadata incomplete. This incompleteness severely hampers any system relying on user-generated metadata.

Metadata heterogeneity: Some of the vocabulary terms introduced by end-users to annotate content locally may not make sense on a larger scale. New vocabulary terms—new attributes or properties used locally by some community—should be related to equivalent vocabulary terms coming from different communities to guarantee interoperability. This is a semantic heterogeneity issue requiring a decentralized integration paradigm, as we have to deal with large and distributed communities of users, which develop without any central authority that could enforce vocabulary terms globally. A similar issue arises when a user makes an explicit reference to a local resource in the collaborative setting: the reference can be totally irrelevant to most of the other users who are not aware of the resource in question.

These two problems are the main reasons why semi-structured metadata are scarce in distributed settings: either metadata are incomplete or they cannot be properly interpreted and are thus simply discarded. An RDF document exhibiting concrete examples of those two problems is shown in Fig. 1.

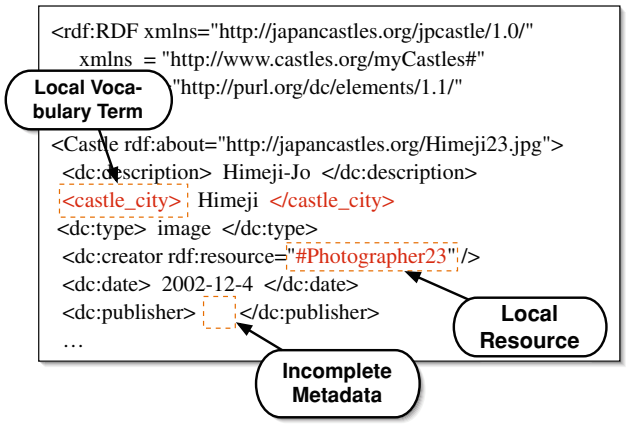

Fig. 1 The two fundamental problems behind semi-structured metadata scarcity in distributed settings: metadata incompleteness caused by missing values, and metadata heterogeneity attributable to local vocabulary terms and local resources

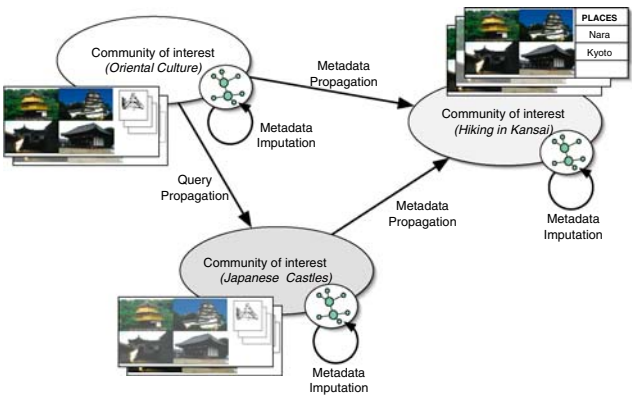

Fig. 2 Sharing semi-structured metadata: metadata incompleteness is mitigated by imputing additional metadata for sets of similar resources inside a community of interest, while metadata heterogeneity is gradually alleviated through pairwise schema mappings

2.3 Opportunities for reducing metadata scarcity collaboratively

In the rest of this contribution, we gradually alleviate metadata scarcity by tackling the two aforementioned problems in a large-scale resource-sharing context. Hence, we focus on methods to solve both metadata incompleteness and metadata heterogeneity for semi-structured metadata attached to the shared resources. Even if sharing semi-structured metadata is intrinsically difficult, we argue that simultaneously sharing both the resources and their associated metadata in large-scale communities opens the door to new opportunities for supporting global search on the shared resources.

Assuming that we can relate shared resources semantically inside a given community of users, e.g., by a low-level analysis of their content or by a semantic analysis of their metadata, metadata can be propagated within the community to reduce metadata scarcity. By taking into account sets of resources shared in a given community, we can thus augment individual metadata by combining local metadata attached to a resource with other metadata originating from similar resources. We call this process *metadata imputation* in Fig. 2. Data imputation is a field aiming at replacing missing values in a data set by some plausible values (see Farhangfar et al. [16] for a recent survey of the field).

By relating resources and metadata coming from different communities of users, we can further enhance the process by creating schema mappings between semantically related communities of users. Schema mappings associate vocabulary terms of one community to related terms coming from another community. They allow the reformulation of a query posed against a given schema into a semantically similar query written in terms of another schema. We refer to this process as *query propagation* in Fig. 2, where straight arrows represent mappings between the schemas of two given communities. Schema mappings can reduce semantic heterogeneity by enabling the propagation of a local query across the whole network of communities by following series of mapping links iteratively.

In this article, we additionally take advantage of schema mappings to propagate existing metadata across semantically heterogeneous communities, and thus to reduce metadata scarcity even further. Metadata imputation is this time contingent on the availability of schema mappings relating the schemas of heterogeneous communities. We refer to this process as *metadata propagation* in Fig. 2.

In turn, metadata that have been propagated through schema mappings can be exploited in order to infer new mappings or verify existing mappings and to increase the accuracy of metadata propagation. This shows a clear interrelation between metadata incompleteness and metadata heterogeneity, as minimizing metadata incompleteness through metadata propagation takes advantage of schema mappings used to minimize semantic heterogeneity, and vice-versa.

In the following, we propose a distributed process to reduce the overall scarcity and heterogeneity of the metadata in an autocatalytic process, where both metadata and mappings get reinforced recursively by putting local metadata into a global community-based context. Taking a global view on the system, we observe that the global semantics are not fixed *a priori*, but evolve as users interact with the system and guide the metadata sharing process by exporting new resources, by adding new metadata, or by providing positive or negative feedback based on the results retrieved for their queries. The way the semantics of the system dynamically evolves is typical of an emergent semantics system [8], where no global semantics is defined *a priori* and where the discovery of the proper interpretation of symbols results from a self-organizing process guided by local interactions.

3 Formal model

The problem we want to tackle can be formally introduced as follows: a large set of autonomous information parties we name *peers* $p \in \mathcal{P}$ store *resources* (e.g., calendar entries, pictures, or video files) $r \in \mathcal{R}_p$ locally. Peers take advantage of *schemas* $S \in \mathcal{S}$ to describe their resources with semi-structured metadata.

Peers using the same schema to describe resources form a *community of interest*. Communities of interest develop independently of our system through social interactions or schema enforcement. They typically result from best practice or standardization efforts, or from communities of users using specialized tools imposing a custom schema (see for example the W3C Incubator Group Report [15] for recent examples of specialized schemas related to the image annotation domain).

Schemas consist of a finite set of vocabulary terms $t \in \mathcal{T}$. We focus on vocabulary terms representing attributes (a.k.a. properties), which invariably exist in one way or another in all the semi-structured metadata formats we have encountered, but classes of resources can be taken into account by our approach as well. In the setting we consider, we assume that the number of shared resources is typically significantly higher than the number of peers, which is itself significantly higher than the number of schemas: $|\mathcal{R}| >> |\mathcal{P}| >> |\mathcal{S}|$.

Peers store the semi-structured metadata attached to their resources as attribute-value pairs. By taking into account the resource each attribute-value pair describes, semi-structured metadata can be seen as ternary relations. We call such relations metadata *statements* (r, t, v). A statement (r, t, v) associates a value v to a local resource r through a vocabulary term (attribute) t. Values v appearing in the statements can either represent literals $l \in \mathcal{L}$ or local resources. For instance, the *dc:description* statement related to the resource whose metadata are shown in Fig. 1 can be written as follows:

$$(http://japancastles.org/Himeji23.jpg,\ dc:description,\ Himeji-Jo).$$

We say that a statement evaluates to $true$ if it exists in one of the databases of the peers, to false otherwise. We suppose at this stage that all annotations are complete, in the sense that for each annotated resource, all vocabulary terms defined in the annotation schemas are associated with a certain value. This also constraints the number of statements attached to each resource based on schemas.

Peers can pose queries locally in order to retrieve specific resources based on vocabulary terms, literals, and other local resources. Queries take the form of conjunctions of triple patterns [31]:

$$r? : (r_1, t_1, v_1), \ldots, (r_n, t_n, v_n)$$

where r_k, t_k, v_k represent variables or (respectively) a local resource, vocabulary term, or value, and $r?$ is a distinguished resource variable appearing in at least one of the triple pattern (r_k, t_k, v_k). Note that joins can be expressed by multiple occurrences of the same variable in that notation. We say that a resource $r_0 \in \mathcal{R}$ is an *answer* to the query q, and write $q \models r_0$, if, when substituted for the distinguished variable, there exists a valuation (i.e., a value assignment) for all other

variables in the conjunction of triple patterns such that the resulting statements all evaluate to $true$.

Now, let us assume that some of the statements are incomplete and that the peers have a means to export resources through some common infrastructure (e.g., the World Wide Web or a Distributed Hash Table). Our goal is to *recontextualize* the local statements in the common infrastructure to support global search capabilities. We create additional statements in such a way that any peer posing a query q against its local schema can retrieve a maximal number of relevant resources $r \mid q \models r$ from the global set of shared resources $\mathcal{R}$ while minimizing false positives and user's involvement under the following restrictions:

Metadata incompleteness: Some values v_k appearing in the statements are replaced by null-values $\perp_k$ introducing incomplete statements $(r_k, t_k, \perp_k)$. Null-values are formally considered as being equivalent to the values they stand for but cannot be distinguished by the peers, which basically consider the values as unknown. For instance, supposing that the *dc:description* from Fig. 1 is left incomplete, the statement can be written as:

$$(http://Himeji23.jpg, dc:description, \perp_{dc:desc}).$$

Metadata heterogeneity: Each local resource and vocabulary term is assigned a set of fixed interpretations r^I from a global domain of interpretations Δ^I with $r^I \subseteq \Delta^I$. These interpretations are used to interrelate semantically similar resources. Arbitrary peers are not aware of such assignments, i.e., they are not aware of the global semantics of the system. We define two resources r_i and r_j as *equivalent*, expressed by $r_i \equiv r_j$, if and only if $r_i^I = r_j^I$. We define that a resource r_i subsumes another resource r_j, expressed by $r_j \sqsubseteq r_i$, if and only if $r_j^I \subseteq r_i^I$. Trueness of statements is relative to the equivalence and subsumption relations, in the sense that if a statement (r, t, v) evaluates to $true$, then all statements $(r', t', v') \mid r' \sqsubseteq r, t' \sqsubseteq t, v' \sqsubseteq v$ also evaluate to $true$. For instance, the following statement:

$$(http://Himeji23.jpg, located\_in\_City, Himeji)$$

evaluates to $true$ if

$$(http://Himeji23.jpg, castle\_city, Himeji)$$

and

$$castle\_city \equiv located\_in\_City\ .$$

Taking advantage of those definitions, we can introduce the notions of metadata completeness and soundness. We say that a set of N statements $\{(r, t_1, v_1), \ldots, (r, t_N, v_N)\}$ pertaining to a resource r is *complete* when $v_i \neq \perp \forall\ v_i$. A set of statements is *sound* if all the statements evaluate to true. We generally assume that our process starts with sets of statements that are sound but incomplete. Our recontextualization process then tries to complete the statements while minimizing the number of unsound statements generated.

3.1 Metadata entropy

We introduce in the following a new metric for capturing metadata scarcity. We call this metric *metadata entropy* as it is similar in nature to the notion of entropy defined in information theory. In our context, metadata entropy either relates to the incompleteness of the metadata statements or to the uncertainty of inferred statements. Keeping track of metadata entropy is important to detect the resources requiring further recontextualization (too many incomplete statements), and to propagate metadata in a meaningful way by associating uncertainty to the metadata that are inferred automatically.

We extend our model to write statement as quadruples $(r, t, \boldsymbol{v}, \boldsymbol{p})$ where $\boldsymbol{v}$ is a list of possible values $v_k \in \boldsymbol{v}$ for the statement and $p_k \in \boldsymbol{p}$ stands for the probability of the statement (r, t, v_k) evaluating to $true$. Using this notation, we can for example write the following:

$$(http://Himeji23.jpg,\ castle\_city,\ (Himeji,\ Kyoto),\ (0.9,\ 0.1))$$

to express that two different cities were related to a given picture.

The *entropy* $H(r, t, \boldsymbol{v}, \boldsymbol{p})$ of a statement measures the degree of uncertainty related to its set of possible values $\boldsymbol{v}$.

We wish metadata entropy $H(\cdot)$ to satisfy the following desirable properties:

1. Metadata entropy $H(\cdot)$ should be a continuous function based on the various probabilities $\boldsymbol{p}$ attached to the values of a statement. It should not, however, depend on the order in which the probabilities or the values are given.
2. Metadata entropy $H(\cdot)$ should evaluate to zero for sound statements.
3. Metadata entropy $H(\cdot)$ should be maximal and evaluate to one when all values attached to a statement are equiprobable, i.e., when no value is more probable than any other.
4. $H(p(X, Y))$ should be equal to $H(p(X))H(p(y))$ for two independent random variables X and Y.

The only function satisfying these four properties [21] is:

$$H(r, t, \boldsymbol{v}, \boldsymbol{p}) = -\sum_{k=1}^{K} p_k log_K(p_k)$$

where K is the number of possible values in $\boldsymbol{v}$. The entropy of all complete statements initially exported by the peers is zero, as they consider a single possible value with a probability of 1 of evaluating to $true$ (i.e., we consider that all statements stored locally at the peers are correct; if some of these statements are created semi-automatically, we can alternatively start with a smaller value $0 < p < 1$). Incomplete statements $(r, t, \perp)$ start with an entropy of one initially, representing an unknown (and potentially infinite) set of identically distributed values. Their entropy will decrease over the course of our recontextualization process as plausible values get discovered trough metadata imputation and propagation.

We define the entropy of a resource as the arithmetic mean of the entropy of its N associated metadata statements:

$$H(r) = \sum_{n=1}^{N} H(r, t_n, \boldsymbol{v_n}, \boldsymbol{p_n}) N^{-1}.$$

A resource with half of its metadata statements left incomplete will thus start with an entropy of 0.5.

4 Recontextualizing semi-structured metadata

We present our general approach for generating additional data to recontextualize shared metadata in large-scale distributed settings below. A specific implementation of this approach is described in Sect. 5 in the context of an image sharing scenario.

The heterogeneity, autonomy, and large number of peers we consider precludes the use of centralized techniques. Traditional integration techniques (e.g., the mediator architecture [34]) are impractical in our context as no global schema can be enforced in heterogeneous and decentralized communities. Classical metadata management techniques (e.g., tableaux reasoning) are not applicable either, due to the lack of shared information (resources, vocabulary terms) and the sheer size of the problem which excludes methods scaling exponentially—or even linearly—with the size of the data [13].

Instead, we propose local, probabilistic heuristics aiming at recontextualizing metadata extracted from a specific source to a decentralized collaborative context. Following a long tradition of providing scalable application-level services on top of an existing physical network, we push the "intelligence" of the approach towards the edge of the network, i.e., perform all complex operations locally at the peers, while only considering simple in-network operations on a shared hash-table. In the following, we suppose that all resources and peers are identified by globally unique identifiers. Our heuristics are based on decentralized data indexing, data imputation, and data integration techniques. We detail below how metadata are exported, how statements are imputed inside a community of interest, how pairwise schema mappings are created to alleviate metadata heterogeneity, how metadata are propagated across communities, and finally how user queries are handled. A high-level illustration of the recontextualization process is given in Fig. 3.

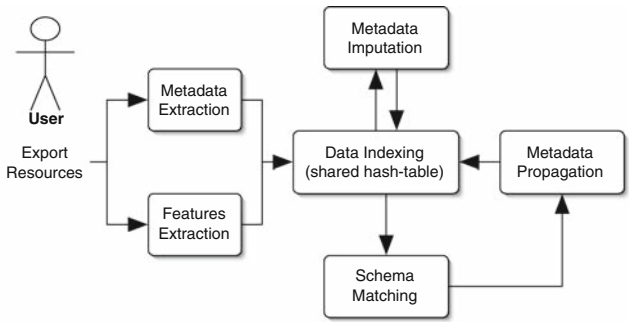

Fig. 3 The metadata recontextualization process: users start by sharing resources, which are indexed in a hash-table along with metadata and content features. Features are used to match resources and impute new metadata, while existing metadata are used to create schema mappings, which in turn are used to propagate metadata from one community to the others

4.1 Exporting local metadata through data indexing

Our recontextualization process starts with the export and proper indexing of the shared resources and their associated metadata. We index the location p_0 of each resource r_0 a peer wants to share in a shared hash-table. We then index all metadata statements $(r_0, t, v, 1)$ pertaining to the resource that has just been indexed. The indexing process continues recursively by indexing all resources r' appearing as values v in the already indexed metadata, and their respective statements $(r', t', v', 1)$. Figure 4 shows an example of the indexing process on a simple RDF graph with recursion depths limited to zero and one. All statements are exported using a common representation (e.g., XML serialization of RDF triples) and are indexed in such a way that they can be efficiently retrieved based on their resource r, term t, or value v. Higher recursion values lead to sharing more information, which can then be used in the rest of the recontextualization process to relate semantically similar resources. On the other hand, higher recursion values also impose higher network traffic and higher load on the shared hash-table.

4.2 Dealing with metadata incompleteness through intra-community metadata imputation

Our objective turns now to determining plausible values for incomplete statements based on sets of related statements. This can be regarded as a data imputation problem [16] where values are missing within a given community of interest. In

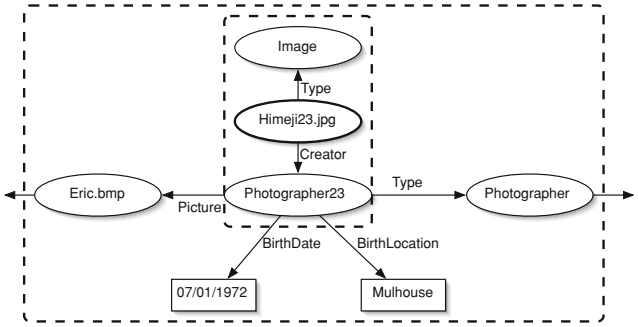

Fig. 4 Indexing resources and statements from an RDF/S graph; the inner and outer boxes correspond to a recursion depth limited to respectively zero and one and starting from the *Himeji23.jpg* resource

our context, metadata are incomplete due to the users' unwillingness to fully annotate the resources. Hence, metadata can be incomplete irrespective of the resource they are attached to or of their actual value (values are missing *completely at random* [16]). We base our imputation process on a K-Nearest Neighbor imputation, which has been shown as being very effective for contexts such as ours [4] and has two distinctive advantages in the present situation: (i) it does not require building a predictive model for each predicate for which a value is missing and (ii) it can be based on a simple index lookup in a shared hash-table [18].

K-Nearest Neighbor imputation is based on a notion of distance between the objects it considers. Capitalizing on several decades of research on content analysis, we generate *feature values* for each resource we have to index. Feature values represent the content of the resource and/or its metadata. Feature values can for example be based on a low-level analysis of the resource (e.g., image analysis) or on an analysis of machine-generated metadata (see next section for some concrete examples). Features should be extracted in such a way that similar resources get closely related feature values, which might or might not be verified in practice and which naturally impacts on the effectiveness of our approach (see Sect. 5.2). Feature extractors might be different for different types of resources (e.g., pictures, text files, etc.). We index each resource r based on its feature value $FV(r)$ in the shared hash-table to be able to retrieve resources with similar feature values. We define the distance used by the imputation process based on those values: $D(r, r') = |FV(r') - FV(r)|$ (see Sect. 5.1 for concrete examples of features and distances).

Algorithm 1 gives a list of the operations undertaken during an imputation round. The imputation can be broken down into three main operations: neighbor selection, value inspection, and value aggregation.

Neighbor selection: For each resource r associated with at least one incomplete statement $(r, t, \perp)$, we search for K similar resources (neighbors) r' in the hash-table, such that r and r' are annotated using the same schema, $D(r, r')$ is

Algorithm 1 Imputation process operations

```
for all resource r to index do
  incompleteStatements = r.getIncompleteStatements()
  if incompleteStatements.count() > 0 then
    /*Neighbor Selection*/
    neighbors = getNearestNeighbors(r, K, τ)
    for all incompleteStatement in incompleteStatements do
      for all neighbor in neighbors do
        /*Value Inspection*/
        plausibleValues = getValuesFromNeighbor(
                          neighbor, incompleteStatement)
        likelihoods = assessLikelihoods(neighbor,
                          incompleteStatement, r)
        listOfValues.add(plausibleValues)
        listOfLikelihoods.add(likelihoods)
      end for
      /*Value Aggregation*/
      aggregate(incompleteStatement,
                          listOfValues, listOfLikelihoods)
    end for
  end if
end for
```

minimal—and below a similarity threshold τ—and $H(r')$ as low as possible. That is, we search for resources coming from the same community of interest that are most similar according to our feature value metric and whose statements are as sound and complete as possible. The exact value of K depends on the context and is typically determined by cross-validation [36] using a sample validation set. When fewer than K similar resources exist in the radius of the similarity threshold τ, abstract resources with incomplete statements $(r_\perp, t, \perp)$ with $D(r, r_\perp) = \tau$ are considered. We introduce abstract resources to explicitly preserve null values when few plausible values are available from the neighborhood of a given resource.

Value inspection: For each incomplete statement $(r, t, \perp)$, we consider the I values v'_{ki} appearing in the corresponding statement $(r'_k, t, \mathbf{v'_k}, \mathbf{p'_k})$ attached to the k^{th} neighbors as a possible value. We set the likelihood l'_{ki} of this value being sound for the incomplete statement under consideration as being proportional to the probability p'_{ki} of the value being itself sound, and inversely proportional to the distance between the two resources:

$$l'_{ki} = p'_{ki} D(r, r'_k)^{-1}.$$

Thus, sound statements or statements coming from very similar instances are systematically preferred.

Value aggregation: Finally, we aggregate the values v'_{ki} and likelihoods l'_{ki} coming from the k chosen neighbors into D distinct values v_d and probabilities p_d, with

$$p_d = \frac{\sum_{\forall k,i | v'_{ki} = v_d} l'_{ki}}{\sum_{k=1}^{K} \sum_{i=1}^{I} l'_{ki}}.$$

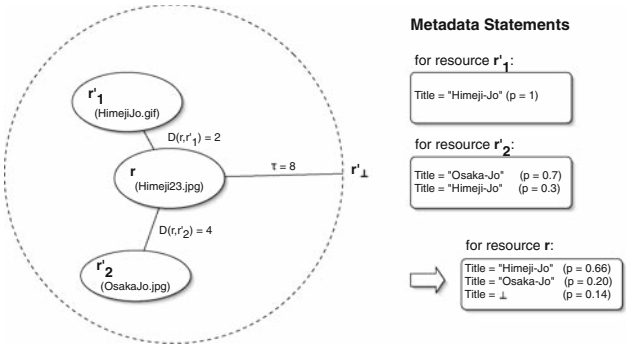

Fig. 5 An example of data imputation: statements coming from two nearby candidate resources r'_1 and r'_2 and an abstract instance $r_\perp$ are combined to complete the statements attached to resource r whose *Title* is missing

Figure 5 shows an example of the imputation process for an incompletely annotated image file r, combining the statements of its two nearest neighbors r' and r''.

Note that a similar imputation process can again take place later on, once the statements have already been recontextualized but new resources have been indexed, for example periodically every T period of time for resources with a high entropy. In a dynamic context and for high values of K, one should additionally avoid storing too many unlikely values by eliminating all values with low probabilities ($p_d < p_{min}$).

4.3 Dealing with metadata heterogeneity through pairwise schema mappings

So far, metadata imputation was limited to a given community of interest. To take advantage of all statements shared by peers outside of a given community, we extend the application of self-organization principles to the creation of mappings relating similar communities of interests. We create mappings between pairs of schemas to link semantically related vocabulary terms. Mappings are used to identify equivalent terms in the data propagation process (see below Sect. 4.4), and to reformulate queries iteratively as in a peer data management system [2,32].

To create schema mappings relating communities, we first have to identify equivalent terms $t' \equiv t''$ from different schemas S' and S''. Various methods can be used to discover those equivalences automatically. Schema matching is an active area of research [28] but is not however the focus of this work. In our large-scale, decentralized context, retrieving all data from the shared hash-table—for instance for the purpose of creating a corpus [23]—would be prohibitively expensive. Methods focusing on selective search queries and piggybacking on other operations should instead be used in order to minimize the overhead on the shared infrastructure.

Fundamentally, we base the semantics of a mapping relating two terms t' and t'' on the likelihood of a statement (r, t', v) being sound, knowing that a similar statement on t'' (r, t'', v) is indeed sound:

$$P(t' \equiv t'') = P((r, t', v) \text{ evaluates to } true \mid (r, t'', v) \text{ evaluates to } true)\ \forall r, v.$$

More pragmatically, we use a simple instance-based schema matching approach piggybacking on the imputation process to approximate this value. We create a new mapping $(t', \equiv, t'', p_\equiv)$ whenever two statements (r', t', v', p') and (r'', t'', v'', p'') on two similar resources r' and r'' with $D(r', r'') < \tau$ with equivalent values $v' \equiv v''$ are discovered during the neighbors selection phase. The process of deciding whether or not two values are equivalent can be based on lexicographical and linguistic analyses (e.g., edit distance between strings, equivalence based on synonyms appearing in a thesaurus). Note that the distance defined by the feature values is here instrumental in creating the mappings, since failing to recognize two resources as being similar invalidates the whole process.

The probability $p_\equiv$ that this relation holds is derived by retrieving analogous statements (r_j, t', v_j, p_j) (r_k, t'', v_k, p_k) from the shared hash-table:

$$p_\equiv = \frac{\sum p_j \ \forall (r_j,t',v_j,p_j),(r_k,t'',v_k,p_k) | D(r_j,r_k)<\tau \wedge v_j \equiv v_k}{\sum p_j \forall (r_j,t',v_j,p_j),(r_k,t'',v_k,p_k) | D(r_j,r_k)<\tau}.$$

The probability is thus computed by counting the number of equivalent values appearing in the instances considered as being similar for the two terms. For instance, indexing two similar images r_1 and r_2 with $D(r_1, r_2) < \tau$ with two statements sharing the same value

$$(r_1,\ located\_in\_City,\ Himeji)$$

and

$$(r_2,\ castle\_city,\ Himeji)$$

would trigger the creation of a mapping between $located\_in\_City$ and $castle\_city$ by retrieving the set of similar resources annotated with either of the two terms and by comparing the values of their statements. The creation of mappings for the other terms appearing in the schemas can be conducted simultaneously.

Incomplete statements (i.e., statements with $v = \perp$) are not taken into account in these computations. Unsound statements (i.e., statements with $p < 1$) are taken into account in this process by weighting their importance with their likeliness (i.e., less likely values v_i with probabilities p_i close to zero will have less impact than more probable values). Note that subsumption mappings can be exported and discovered in an identical manner by taking into account subsumption relations $\sqsubseteq$ in place of the equivalence relations above.

In highly dynamic environments where new statements are inserted on a continuous basis, recomputing $p_\equiv$ each time a new pair of potentially related terms is discovered would be

expensive. In such situations, the decision to recompute $p_{\equiv}$ can be based on the number of times the triggering condition is observed by a particular peer, or on an analysis of the graph of schemas and mappings determining whether or not more mappings would be useful for reformulation purposes [9].

4.4 Dealing with metadata incompleteness through inter-community metadata propagation

We can now extend the imputation process by propagating metadata across different communities of interest following schema mappings. The process is similar to the imputation process confined to a single community of interest (see Sect. 4.2), but takes this time into consideration all neighbors annotated with equivalent schemas related through mappings.

For a resource r and given an incomplete statement $(r, t, \perp)$, K neighbors r'_k are considered such that $D(r, r'_k)$ is minimal and below τ, $H(r'_k)$ is as low as possible, and based on the existence of statements $(r'_k, t'_k, \mathbf{v'_k}, \mathbf{p'_k})$ such that t and t'_k are either identical or related through a mapping $(t, \equiv, t'_k, p_{k\equiv})$. The likelihoods l'_{ki} attached to the values is then weighted with $p_{k\equiv}$ to account for the fact that the mapping is in itself uncertain:

$$l'_{ki} = p'_{ki} D(r, r'_k)^{-1} p_{k\equiv}.$$

As an example, suppose that the resource r'_1 in Fig. 5 is annotated with a different schema considering an attribute *legend* similar to $title$, with $(title, \equiv, legend, 0.5)$. The uncertain mapping would then reduce the significance of r_1's contribution, lowering the likelihood of $Himeji - Jo$ to 0.52.

Note that a value can be propagated iteratively across series of communities of interest in that manner, and that propagated values can in turn bootstrap the creation of new schema mappings.

4.5 Possible answers and user feedback

User queries $r?: (r_1, t_1, v_1), \ldots, (r_n, t_n, v_n)$ can be resolved by iterative lookup on the shared hash-table: for each triple pattern in the query, candidate triples are retrieved by looking-up one of the constant terms of the triple pattern in the shared hash-table [10]. Answers to the query are then obtained by combining the candidate triples. In addition to the certain answers obtained in that way, *possible* answers [11] are generated by reformulating queries following (probabilistic) schema mappings to query distant communities of interest:

$$r'?: (r_1, t'_1, v_1), \ldots, (r_n, t'_n, v_n) \\ \mid (\exists S_j \mid t'_1, \ldots, t'_n \in S_j) \wedge t'_1 \equiv t_1, \ldots, t'_n \equiv t_n$$

and by taking into account the probabilities attached to the values of the entropic statements generated by the metadata imputation and propagation processes. The resulting answers can be ranked with respect to their likelihood to present the most likely results first to the user. Optionally, resources with a high entropy (i.e., resources with many incomplete or unsound statements) can at this stage be proposed to the user in order to take advantage of his feedback to classify those highly uncertain resources and to bootstrap a new data imputation round.

5 PicShark: sharing annotated pictures in the large

To demonstrate the viability of our metadata recontextualization strategies, we developed a system called *PicShark*. PicShark is an application built on top of a semantic overlay network allowing global searches on shared pictures annotated with incomplete, local and semi-structured metadata. We consider PicShark as a concrete instantiation of the imputation principles described above. PicShark concentrates on the image annotation domain, but it would be straightforward to extend our application to other resources (e.g., textual documents, videos, or music files) by taking advantage of different feature domains and defining new distances.

Our approach extends the principle of data independence [19] by separating a logical layer—a semantic overlay managing structured metadata and schemas—from a networking layer consisting of a structured Peer-to-Peer overlay used for efficient routing of messages (see Fig. 6). The networking layer is used to implement various functions at the logical layer, including query resolution, information imputation, and information integration.

We use P-Grid [1] as a substrate for storing all shared information in a Distributed Hash-Table (DHT). Indexing of statements is handled by GridVine [10]. GridVine provides efficient mechanisms for storing triples (or quadruples in our case) in a decentralized way, and facilitates efficient resolu-

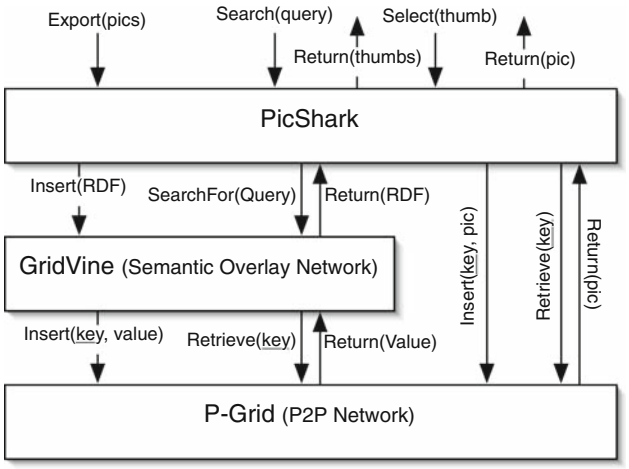

Fig. 6 The PicShark architecture: PicShark uses P-Grid to store shared resources and GridVine to share semi-structured metadata

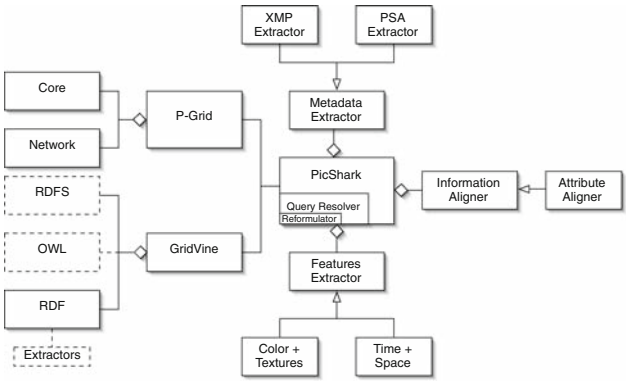

Fig. 7 The PicShark components: PicShark uses *metadata extractors* to syntactically export all metadata using a common format in the shared hash-table, *aligners* to align schemas on a semantic level, and *feature extractors* to extract low-level features representing the images

tion of conjunctive queries in $\mathcal{O}(log(n))$ messages, where n is the number of peers in the system.

On top of this architecture, PicShark takes care of fostering global semantic interoperability by recontextualizing local statements exported to the P2P network. Users can export sets of local pictures through the *Export(pics)* method and search for pictures by specifying conjunctive queries against their local schemas.

Figure 7 gives an overview of the various components used in PicShark. Data indexing takes advantage of *metadata extractors* (see below) to syntactically align the statements before sharing them through GridVine. Creation of mappings is handled by *aligners*, while *feature extractors* are used to extract the feature vectors used to relate similar images.

5.1 Information extraction in PicShark

PicShark uses *metadata extractors* to extract local metadata from the images and to syntactically align the metadata to a common representation. PicShark uses RDF/S as a common syntax, and converts all supported metadata formats to this representation. The application currently supports two very different semi-structured metadata formats: PSA, which is a hierarchical, proprietary format used by Photoshop Album[3] and XMP, which is a standard based on RDF/S. Both standards are extensible and let users define new vocabulary terms to annotate their pictures. The *PSA Extractor* extracts semi-structured statements and vocabulary terms from the local relational database used by Photoshop Album to store all metadata, while its XMP counterpart extracts statements and vocabulary terms from the payload of the pictures. The extractors generate all missing GUIDs (for local vocabulary terms and pictures), index statements using GridVine and images using P-Grid directly. All statement values are stored as strings. The system directly compares stemmed versions of the strings to determine whether or not two values are equivalent.

Features can be extracted from the images either by a low-level analysis based on sixty texture and color moments, or by the extraction of spatial and temporal metadata from the images. With time-stamps directly embedded into most digital images and with the proliferation of GPS devices and localization services (such as ZoneTag[4]), we believe that the combination of both temporal and spatial information represents a new and computationally inexpensive way of identifying related images (see also below Sect. 5.2 for a discussion on that topic). Based on the extracted features, similarity search retrieving closely related resources during the imputation process can be implemented efficiently in our setting by using locality-sensitive hashing [18] at the networking layer. Distances for both feature spaces are defined as standard Euclidian distances (respectively for sixty and two dimensions).

5.2 Performance evaluation

Evaluating the performance of a system like PicShark is intrinsically difficult for several reasons. PicShark is (to the best of our knowledge) the first application taking advantage of semi-structured, heterogeneous, and incomplete metadata statements. As semi-structured statements from popular image organization applications such as Photoshop Album or Extensis Portfolio[5] are kept in local databases and are not searchable on the Web, constituting a realistic and sufficiently large data set is currently difficult. Using tag collections from popular tagging portals such as Flickr is impossible as well, as the tags are very noisy and totally unstructured. Moreover, recontextualization is a highly recursive, distributed and parallel process, such that getting a clear idea of the ins and outs of the operation is difficult for large data sets or numerous peers.

In the following, we describe a set of controlled experiments pertaining to a set of three hundred photos[6], which were manually annotated using Adobe Photoshop Album Starter Edition 3. The set of photos is divided into three subsets, each taken by a different individual during a common trip to Japan. All sets were annotated using Photoshop Album. The first two subsets use the same schema, while the third subset was annotated using a different—but semantically related—schema. Schemas were designed by the photographers and consist of about ten attributes each. Temporal information was directly taken from the time-stamp embed-

[3] http://www.adobe.com/products/photoshopalbum/starter.html.

[4] http://research.yahoo.com/zonetag/.

[5] http://www.extensis.com/.

[6] both photos and semi-structured metadata are available for download at http://lsirwww.epfl.ch/PicShark/.

ded by the cameras, while spatial information (i.e., GPS coordinates) was added manually to each picture.

5.2.1 Intra-community imputation

This first experiment focuses on analyzing results pertaining to metadata imputation in a given community of interest. We start by exporting the first two subsets of 100 images each, along with their metadata. We randomly drop each statement—except spatial and temporal information, which are always preserved—with a probability $pMissing$ to simulate metadata scarcity. We then recontextualize the 100 images from the first subset one by one using images and statements from the second subset to simulate intra-community recontextualization (remember that both subsets use the same schema). We alternatively base the imputation process on either low-level features or spatial and temporal metadata. As our image set is pretty homogeneous, we set $\tau = \infty$ and $K = 2$, i.e., we always take the two nearest neighbors to recontextualize a given picture.

Figure 8 shows the evolution of the total metadata entropy pertaining to the first subset of photos during the recontextualization process, for various values of $pMissing$ ranging from 20% to 80%. The figure gives a normalized value of the total entropy (the absolute entropies start at 82, 166, and 329 for $pMissing = 20$, 40, and 80% respectively). Total entropy offers in our context a finer granularity to analyze the process than, say, a standard recall metric, which would be inadequate to capture the distribution of values attached to the propagated statements. The curves depicted on Fig. 8 represent average results obtained over 10 consecutive runs. Note that the results are pretty stable: the standard deviation never exceeds 10% of the absolute value. The entropy—and thus, the uncertainty on the set of images—decreases as more and more pictures get recontextualized. The imputation process based on spatial and temporal values ($S+T$) is slightly better than the process based on low-level features (LL) at finding images with very related statements. For high $pMissing$ values, many values are missing and fewer metadata statements get propagated.

The impact of the nearest-neighbor search is best illustrated by Figs. 9 and 10, which respectively depict the aggregated probability for the sound and unsound metadata generated by the system. We call aggregated probability the sum of the probabilities attached to propagated metadata (propagated metadata with $\perp$ values are not taken into account). Note that propagating metadata usually decreases the total entropy of the system, except when highly uncertain metadata are generated (e.g., when a $\perp$ value is replaced by two

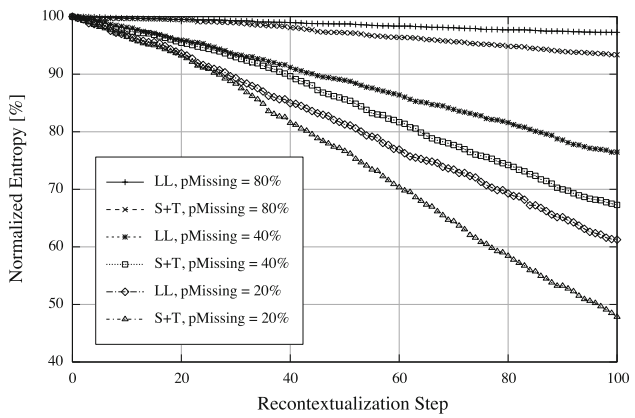

Fig. 8 Normalized total entropy pertaining to the first subset of images, for metadata missing with various probabilities $pMissing$; at each step, we recontextualize one of the 100 images from the first subset with its two nearest neighbors from the second subset, using either low-level features (LL) or spatial and temporal information ($S + T$)

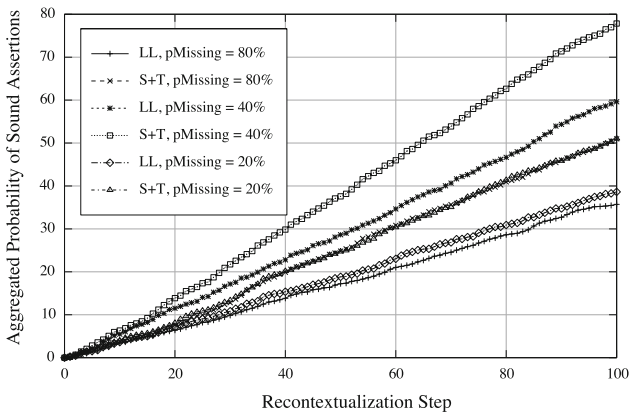

Fig. 9 Aggregated probability of the sound statements generated by the system, for metadata missing with various probabilities $pMissing$; at each step, we recontextualize one of the 100 images from the first subset with its two nearest neighbors from the second subset, using either low-level features (LL) or spatial and temporal information ($S + T$)

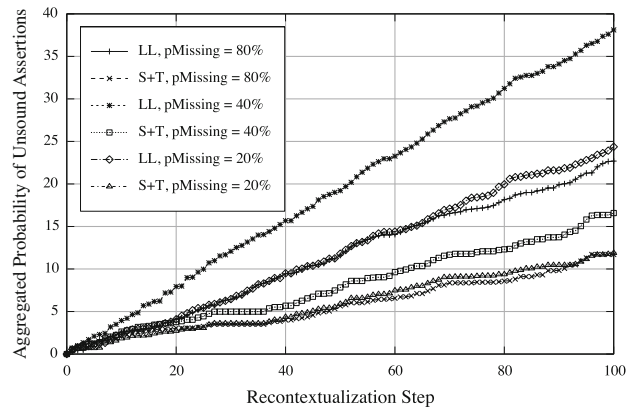

Fig. 10 Aggregated probability of the unsound statements generated by the system, for metadata missing with various probabilities $pMissing$; at each step, we recontextualize one of the 100 images from the first subset with its two nearest neighbors from the second subset, using either low-level features (LL) or spatial and temporal information ($S + T$)

generated values with 50% probability each). $S + T$ is systematically better than LL at finding good neighbors, as it always generates more sound and less unsound statements than LL. This is not surprising, as finding similar photos based on color and texture moments only is known to be a challenging problem in general. $S+T$ generates high-quality metadata that are sound more than 80% of the time. On the other hand, $S + T$ is often wrong when propagating metadata about people appearing on the pictures: here, spatial and temporal information is typically not sufficient and a combination of both $S + T$ and LL would probably be more effective.

In absolute terms, more statements are propagated for $pMissing = 40\%$. For $pMissing = 20\%$, few metadata are propagated (few values are missing), while for $pMissing = 80\%$, few values are available for propagation initially.

5.2.2 Inter-community propagation

In the second part of the experiment, we continue the recontextualization process started above and further recontextualize the 100 photos coming from the first subset with 100 photos coming from the third subset annotated with a different schema. In that way, we simulate the creation of mappings and the propagation of metadata across different communities of interest.

First, the third set of images and their related metadata are exported. We do not drop metadata in the third set, thus simulating a large set of metadata encoded according to different schemas. Schema mappings are created between the two schemas using the instance-based method described in the preceding section. Once the mappings are created, we further recontextualize each of the 100 images of the first set with their two closest-neighbors from the third set. We only use $S + T$ this time, as LL systematically yields inferior results as for the intra-community recontextualization step described above. Figure 11 gives the evolution of the normalized entropy for the first set of images. More uncertain metadata are propagated than for the previous case due to the mappings, which were generated totally automatically based on the values of the statements and are uncertain in this case. Images with a high entropy (e.g., for $pMissing = 80\%$), however, benefit a lot from this second recontextualization round, since their statements were still largely incomplete after the first recontextualization round and since all statements from the third set are complete.

Figure 12 shows the aggregated probability of the sound statements generated during this second round of recontextualization. Unsound statements follow a similar trend, but never represent more than 20% of the generated statements. At the end of our recontextualization process and depending on the value of $pMissing$, 60% to 75% of the initial entropy

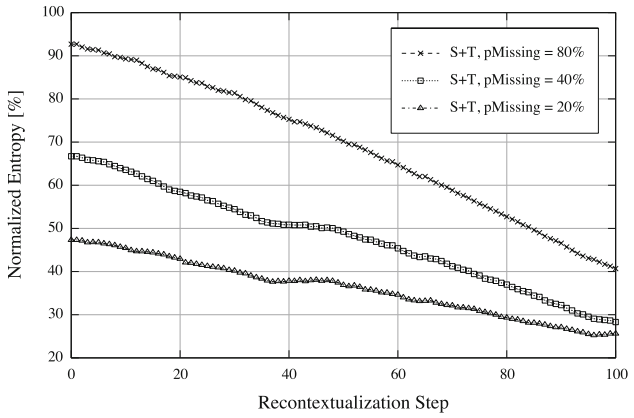

Fig. 11 Normalized total entropy pertaining to the first subset of images, for metadata missing with various probabilities $pMissing$; at each step, we recontextualize an image from the first subset of images with its two nearest neighbors from the third subset, based on spatial and temporal information ($S + T$)

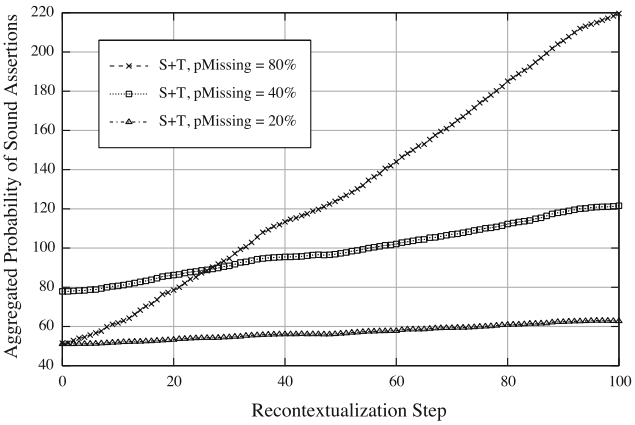

Fig. 12 Aggregated probability of the sound statements generated by the system, for metadata missing with various probabilities $pMissing$; at each step, we recontextualize an image from the first subset of images with its two nearest neighbors from the third subset, based on spatial and temporal information ($S + T$)

of the system induced by incomplete metadata has been alleviated. Most statements contain now entropic metadata that are sound in their majority (less than 20% of the propagated statement are unsound on average with $S+T$). Also, schema mappings relating the two communities of interest have been created automatically. Thus, we are now able to query the system and retrieve relevant images from both communities, while this was totally impossible before the recontextualization process because of the heterogeneity and the lack of metadata.

6 Related work

To the best of our knowledge, our approach is the first one to tackle metadata scarcity in distributed settings. We place

our work at the confines of decentralized data integration, tagging systems, personal information management, and data imputation techniques.

The way we propagate queries in PicShark is typical of a new type of large-scale data integration infrastructures named Peer Data Management Systems (PDMSs). PDMSs integrate data by replacing the centralized mediator by an unstructured network of pairwise schema mappings and by reformulating the queries iteratively from one schema to the others. The complexity of query reformulation in PDMSs is investigated in the context of the Piazza [32] project, while the Hyperion [2] system proposes to use mappings at both the instance and at the schema levels to reformulate queries. PicShark is the first PDMS taking into account probabilistic tuples and supporting the propagation of instances from one database to the others through schema mappings.

Tagging systems, allowing communities of users to add unstructured text labels to resources shared online, are very popular today. While they represent an effective method for gathering large amounts of metadata, the tags they take advantage of represent unstructured information and therefore are difficult to process automatically. Thus, several recent research efforts concentrate on extracting additional structured information from unstructured tags. Rattenbury et al. [29] apply burst-analysis techniques to extract event and place semantics from image tags based on their usage patterns. Schmitz et al. [30] use a subsumption-based model to extract ontologies from Flickr tags, while the ELSABer system [22] organizes tags in hierarchies in order to enable semantic browsing. Wu et al. [35] study the emergence of semantics from tags, resources, and users co-occurrences. In a similar context, Aurnhammer et al. [3] proposed an emergent semantics approach to retrieve images based on collaborative tagging. All these initiatives recognize the importance of semi-structured metadata to improve search in large-scale settings, but non of them tackles scarcity or heterogeneity of metadata.

Similar to PicShark, personal information management systems try to organize data originating from user desktops. Haystack [20] is an information management system, which uses extractors and lets non-technical users teach the application how to extract Semantic Web content to generate RDF triples from various sources. Gnowsis [17] is a semantic desktop where semantic information is collected from different applications on the desktop and integrated with information coming from external tagging portals. Semantic annotations are either extracted or derived from user interaction. The Semex System [12] is a platform for personal information management that reconciles heterogeneous references to the same real-world object using context information and similarity values computed from related entities. The system leverages on previous mappings provided by the users and on object and association databases to foster interoperability. Reconciliation of data was also recently revisited in the context of the ORCHESTRA [33] project. In ORCHESTRA, participants publish their data on an ad hoc basis and simultaneously reconcile updates with those published by others. P-Tag [7] is a system which automatically generates personalized tags for annotating web pages, based on the data residing on the user's personal desktop. Closer to our work, Naaman et al. [26] add identity tags to photos in local photo collections, based on time and location of photographs and co-occurrence of people. Contrary to PicShark, none of these approaches addresses data scarcity or takes advantage of communities of users to collaboratively augment the data that is shared.

Data imputation, finally, denotes techniques aiming at replacing missing values in a data set by some plausible values (see Farhangfar et al. [16] for a recent survey of the field).

7 Conclusions

With the rapid emergence of socially driven applications on the Web, self-organization principles have once again proven their practicability and scalability: through Technorati Ranking[7], Flickr Interestingness[8] or del.icio.us recommendations[9], an ever-increasing portion of the Web self-organizes around end-user input. While most efforts concentrate on unstructured metadata (i.e., keyword) management, we proposed in the article to tackle the problem of organizing structured, heterogeneous metadata in large-scale settings. We advocated a decentralized, community-based and imperfect (in terms of soundness and completeness) way of augmenting semi-structured metadata through self-organizing assertions. Our PicShark system aims at creating metadata automatically by using intra and inter-domain propagation of entropic statements and schema alignment through decentralized instance-based schema matching.

PicShark represents a first proof-of-concept of the applicability of self-organization principles to the organization of semi-structured, heterogeneous and partially annotated content in large-scale settings. We showed in our experiments how incomplete metadata could be enhanced collaboratively using our approach. To the best of our knowledge, PicShark is currently the only system capable of using incomplete and heterogeneous data sets such as the one we used to foster global, structured search capabilities automatically. This first implementation effort opens the door to many technical refinements. As future work, we plan to improve our imputation process to include personalized and fuzzy classification rules to relate semantically similar content. Also, we

[7] http://www.technorati.com/.

[8] http://www.flickr.com/.

[9] http://del.icio.us/.

intend to analyze the system churn—in terms of total entropy, user feedback, and recently indexed information—in order to determine the optimal scheduling of recontextualization and schema matching rounds. Finally, we want to improve the deployability of our application in order to test our approach in situ on large and heterogeneous communities of real users, and are currently launching an initiative jointly with an art center in that context.

References

1. Aberer, K., Cudré-Mauroux, P., Datta, A., Despotovic, Z., Hauswirth, M., Punceva, M., Schmidt, R.: P-grid: A self-organizing structured p2p system. ACM SIGMOD Rec. **32**(3), 29–33 (2003)
2. Arenas, M., Kantere, V., Kementsietsidis, A., Kiringa, I., Miller, R.J., Mylopoulos, J.: The Hyperion project: from data integration to data coordination. ACM SIGMOD Rec. **32**(3), 53–58 (2003)
3. Aurnhammer, M., Hanappe, P., Steels, L.: Integrating collaborative tagging and emergent semantics for image retrieval. In: Collaborative Web Tagging Workshop (2006)
4. Batista, G., Monard, M.C.: An analysis of four missing data treatment methods for supervised learning. Appl. Artif. Intell. **17**(5–6), 519–533 (2003)
5. Boag, S., Chamberlin, D., Fernández, M.F., Florescu D., Robie J., Siméon, J. (ed.): XQuery 1.0: An XML Query Language. W3C Candidate Recommendation, June (2006) http://www.w3.org/TR/xquery/
6. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau F.: (Ed.). Extensible Markup Language (XML) 1.0. W3C Recommendation, February (2004) http://www.w3.org/TR/REC-xml/
7. Chirita, P.-A., Costache, S., Nejdl, W., Handschuh, S.: P-tag: large scale automatic generation of personalized annotation tags for the web. In: International World Wide Web Conference (WWW) (2007)
8. Cudré-Mauroux, P.: Emergent semantics: Interoperability in large-scale decentralized information systems. CRC Press, LLC (2008)
9. Cudré-Mauroux, P., Aberer, K.: a necessary condition for semantic interoperability in the large. In: Ontologies, DataBases, and Applications of Semantics for Large Scale Information Systems (ODBASE) (2004)
10. Cudré-Mauroux, P.: Suchit Agarwal, and Karl Aberer. Gridvine: An infrastructure for peer information management. IEEE Internet Comput. **11**(5), 36–44 (2007)
11. Dalvi, N.N., Suciu, D.: Efficient query evaluation on probabilistic databases. In: International Conference on Very Large Data Bases (VLDB) (2004)
12. Dong, X., Halevy, A.Y.: A platform for personal information management and integration. In: CIDR (2005)
13. Donini, F.M.: Complexity of reasoning. In: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, London (2003)
14. Dumais, S.T., Cutrell, E., Cadiz, J.J., Jancke, G., Sarin, R., Robbins, D.C.: Stuff I've seen: a system for personal information retrieval and re-use. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2003)
15. Troncy, R., et al. (ed.) Image annotation on the semantic web. W3C Incubator Group Report, August (2007) http://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/.
16. Farhangfar, A., Kurgan, L.A., Pedrycz, W.: Experimental analysis of methods for imputation of missing values in databases. Intell. Comput. Theory Appl. II. **5421**(1), 172–182 (2004)
17. Fluit, C., Horak, B., Grimnes, G.A., Dengel, A., Nadeem, D., Sauermann, L., Heim, D., Kiesel, M.: Semantic desktop 2.0: The gnowsis experience. In: International Semantic Web Conference (ISWC) (2006)
18. Haghani, P., Michel, S., Cudré-Mauroux, P., Aberer, K.: LSH at large—distributed KNN search in high dimensions. In: International Workshop on Web and Databases (WebDB) (2008)
19. Hellerstein, J.M.: Toward network data independence. ACM SIGMOD Rec. **32**(3), 34–40 (2003)
20. Karger, D.R., Bakshi, K., Huynh, D., Quan, D., Sinha, V.: Haystack: A general-purpose information management tool for end users based on semistructured data. In: Biennial Conference on Innovative Data Systems Research (CIDR) (2005)
21. Khinchin, A.I.: Mathematical Foundations of Information Theory. Dover Publications, Inc., New York (1957)
22. Li, R., Bao, S., Fei, B., Su, Z., Yu, Y.: Towards effective browsing of large scale social annotations. In: Proceedings of the WWW 2007, pp. 943–952 (2007)
23. Madhavan, J., Bernstein, P.A., Doan, A., Alon Halevy, Y.: Corpus-based schema matching. In: International Conference on Data Engineering (ICDE) (2005)
24. Manola, F., Miller, E. (ed.): RDF Primer. W3C recommendation, February (2004) http://www.w3.org/TR/rdf-primer/
25. McGuinness, D.L., van Harmelen, F.: (Ed.). OWL web ontology language overview. W3C Recommendation, February (2004) http://www.w3.org/TR/owl-features/
26. Naaman, M., Yeh, R.B., Garcia-Molina, H., Paepcke, A.: Leveraging context to resolve identity in photo albums. In: ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL) (2005)
27. Prud'hommeaux, E., Seaborne van Harmelen, A. (ed.): SPARQL Query Language for RDF. W3C Candidate Recommendation, April (2006) http://www.w3.org/TR/rdf-sparql-query/
28. Rahm, E., Bernstein, P.: A survey of approaches to automatic schema matching. VLDB J. **10**(4), 334–350 (2001)
29. Rattenbury, T., Good, N., Naaman, M.: Towards automatic extraction of event and place semantics from flickr tags. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2007)
30. Schmitz, P.: Inducing ontology from flickr tags. In: Collaborative Web Tagging Workshop Edinburgh, Scotland, (2006)
31. Seaborne, A.: RDQL - A Query Language for RDF. W3C Member Submission, 2004. http://www.w3.org/Submission/RDQL/
32. Tatarinov, I., Ives, Z., Madhavan, J., Halevy, A., Suciu, D., Dalvi, N., Dong, X., Kadiyaska, Y., Miklau, G., Mork, P.: The Piazza Peer Data Management Project. ACM SIGMOD Rec. **32**(3), 47–52 (2003)
33. Taylor, N.E., Ives, Z.G.: Reconciling while tolerating disagreement in collaborative data sharing. In: SIGMOD Conference (2006)
34. Wiederhold, G.: Mediators in the Architecture of Future Information Systems. IEEE Comput. **25**(3), 38–49 (1992)
35. Wu, X., Zhang, L., Yu, Y.: Exploring social annotations for the semantic web. In International World Wide Web Conference (WWW) New York, New York (2006)
36. Yang, Y., Ault, T., Pierce, T.: Combining multiple learning strategies for effective cross validation. In: International Conference on Machine Learning (ICML) (2000)