J Data Semant (2016) 5:211–228
DOI 10.1007/s13740-016-0063-6

ORIGINAL ARTICLE

Providing Insight into Data Source Topics

Sonia Bergamaschi[1] · **Davide Ferrari**[2] · **Francesco Guerra**[1] · **Giovanni Simonini**[1] · **Yannis Velegrakis**[3]

Received: 15 February 2015 / Revised: 5 April 2016 / Accepted: 18 April 2016 / Published online: 4 May 2016

Abstract A fundamental service for the exploitation of the modern large data sources that are available online is the ability to identify the topics of the data that they contain. Unfortunately, the heterogeneity and lack of centralized control makes it difficult to identify the topics directly from the actual values used in the sources. We present an approach that generates signatures of sources that are matched against a reference vocabulary of concepts through the respective signature to generate a description of the topics of the source in terms of this reference vocabulary. The reference vocabulary may be provided ready, may be created manually, or may be created by applying our signature-generated algorithm over a well-curated data source with a clear identification of topics. In our particular case, we have used DBpedia for the creation of the vocabulary, since it is one of the largest known collections of entities and concepts. The signatures are generated by exploiting the entropy and the mutual information of the attributes of the sources to generate semantic identifiers of the various attributes, which combined together form a unique signature of the concepts (i.e. the topics) of the source. The generation of the identifiers is based on the entropy of the values of the attributes; thus, they are independent of naming heterogeneity of attributes or tables. Although the use of traditional information-theoretical quantities such as entropy and mutual information is not new, they may become untrustworthy due to their sensitivity to overfitting, and require an equal number of samples used to construct the reference vocabulary. To overcome these limitations, we normalize and use pseudo-additive entropy measures, which automatically downweight the role of vocabulary items and property values with very low frequencies, resulting in a more stable solution than the traditional counterparts. We have materialized our theory in a system called *WHATSIT* and we experimentally demonstrate its effectiveness.

1 Introduction

An increasing number of structured data sources is nowadays becoming available online. A characteristic trend is the one of *open data* in which public and private organizations are making their structured data freely available online, typically in RDF format. Traditional search engines are designed to operate only on document content, which means that the data of these structured data sources are left out of their exploitation sphere.

Identifying the topics of the online structured data sources is of paramount importance, since it will allow them to be indexed by search engines and the references to their content be included in query answers. Indexing structured data is fundamentally different and significantly more challenging than indexing flat, unstructured documents, since the structure plays an important role in the semantics of each value in the data. To overcome this limitation, structured data portals like CTAN[1] have been developed. They play a role similar to that played by search engines for Web documents, but are based on metadata information that has been explicitly provided by the owners of the datasets. Providing this metadata information is a tedious and error-prone task that can also

✉ Francesco Guerra
francesco.guerra@unimore.it

1 DIEF-University of Modena and Reggio Emilia, Modena, Italy

2 The University of Melbourne, Melbourne, Australia

3 DISI-University of Trento, Trento, Italy

[1] http://ckan.org.

introduce bias. Furthermore, it is an approach that does not scale easily. Thus, there is a need for a way to identify the topics of the sources in a way that is automatic, can scale at large, and is also robust to the heterogeneity that is typically observed across independently developed data sources.

The ability to automatically identify the topics of the structured data sources will make it equally important to the static Web pages, promote further the idea of open data, contribute significantly towards the materialization of the "Web of Data" (as opposed to the "Web of Documents"), and will offer countless opportunities for large-scale data analytics [9]. They are not few the efforts to exploit the part of the Web that is hidden behind Web forms and compassable links, i.e. the so-called *hidden Web* [24]. Other efforts have focused on the extraction of structured data which is already available in html pages (e.g. the Web tables [1]) and not on sources that expose their datasets directly.

Furthermore, although there have been efforts to add semantics to Web pages (by means of microdata, rdfa and microformats) and to online structured data sources, these works have become restricted to the value level, ignoring the important semantic information that the structure and the schema in general can offer [15].

We advocate that it is possible to recognize the concepts used in a data source by exploiting some statistical properties of the values. In particular, we claim that the entropy of the set of values of an attribute in the data can be used as an identifier of the specific attribute. The advantage of the entropy is that it does not depend on the actual values of the attribute, but on their distributions in a specific domain. We claim, and our experiments confirm, that typically this distribution does not depend on a specific source, but is a feature of the domain represented by the attribute. All these are making entropy a very promising identifier of what an attribute is describing. The identifiers of the different attributes in the data can be combined together to form a signature of the concepts represented in the source. It is of course possible that two different attributes have the same entropy value. This means that the entropy is not always a unique identifier for an attribute. However, the fact that we consider the combination of the entropies of the individual attributes in the way that they are modelled in the source reduces significantly the chances that a concept would be mistakenly taken for another. The concept signatures in a source collectively can form a signature of the contents of the source.

The representation of the concepts in a source is definitely not a complete representation. This is a consequence to the heterogeneity that is naturally embedded in every source and depends highly on the data of interest upon which the source has been created. For this reason, and to obtain a more accurate semantic representation of the concepts in the source, we match their signatures against a vocabulary of signatures that is more complete and use these signatures in the vocabulary as representations of the source.

The use of statistical properties for recognizing semantically equivalent or related properties has been exploited with success in the past and in particular in the field of schema matching [13]. Inspired by that work, we extend the idea and apply it in the area of source topic detection. Existing approaches have so far been based on the "classical" values of the Shannon entropy. Our experiments have shown that these metrics need to be normalized and may generate unstable results in real environments. This is because the frequency distribution of repeated values in real property domains is typically right skewed, i.e. only few values are significantly repeated, which is problematic because the entropy and mutual information are typically sensitive to regions corresponding to small probabilities, and also because their range depends on the cardinality of the attribute domain (i.e. the number of unique values). This makes the classical values of entropy and mutual information not usable for our case, and we instead use pseudo-additive versions of the classic Shannon entropy.

The specific contributions of this work can be summarized as follows:

- We propose a technique for modelling source topics that is independent of the names of the source structures, hence, independent of many of the complications that structural heterogeneity introduces. The technique uses some statistic metric to generate identifiers of the various attributes that combined together form signatures of the concepts mentioned in the data source.
- We illustrate that the traditional entropy measure is very sensitive and not suitable for many practical cases, so we introduce and use a pseudo-additive versions of it.
- We use our technique to effectively create a required vocabulary of concept signatures based on the information provided by DBpedia[2].
- We use a matching algorithm to match the generated concept signatures of the source to signatures of concepts in the reference vocabulary.
- We describe the materialization of our theory into a system called *WHATSIT*.
- We provide an extensive set of experimental evaluation with real data that illustrates the effectiveness of our approach and discuss our interesting findings.

The remainder of the paper is organized as follows: Sect. 2 introduces formally the problem. Section 4 describes our variations to entropy and mutual information. Section 5 contains our approach and its materialization in the *WHATSIT* system. Section 7 provides the results of our experimental

[2] http://dbpedia.org.

evaluation and Sect. 8 positions our approach in relationship to the related works.

2 Preliminaries

We consider an entity-based data model. We assume the existence of an atomic domain $\mathcal{A}$. Of course, there may be more than one atomic domain such as *String*, *Integer* and *Date*, but for simplicity we consider here only one. We also assume the existence of an infinite set $\mathcal{C}$ of class names and an infinite set $\mathcal{N}$ of property names.

A *property* is a pair $\langle p, d\rangle$, where $p \in \mathcal{N}$ and $d \in \mathcal{A} \cup \mathcal{C}$. The part p is referred to as the name of the property and the part d as the *domain*. We will denote as $\mathcal{P}$ the set of all possible properties. A *class* is a pair $\langle c, P\rangle$, where $c \in \mathcal{C}$, and $P \subset \mathcal{P}$ and is finite.

A data source schema is a finite set of classes C, such that, for every $\langle c, P\rangle$, $\langle c', P'\rangle \in C$, with $\langle c, P\rangle \neq \langle c', P'\rangle \in C$, $c \neq c'$. In short, it means that in a data source, there cannot be two classes with the same class name. For this reason, we can consider a class and its class name as equivalent and by abuse of notation we could write c. Furthermore, if $\langle c, P\rangle \in C$, and $\langle p, d\rangle \in P$, either $d = \mathcal{A}$ or $d \in C$, which means that a property can have an atomic domain or one of the classes of the data source schema.

Our model is generic enough to model the popular relational and RDF schemas. A relational database can be modelled by creating a class for every relational table, in which the class name is the name of the table and the set of properties names consists of one property for each table attribute. The name and the domain of each property is the name and domain of the respective table attribute.

The schema of an RDF database can be modelled in a similar way. A class is created for every RDF class. The name of the class is the name of the RDF class, and the set of properties contain one property for every RDF property that has as a subject the specific RDF class. The name of the property is the predicate of the respective RDF property, while the domain is the object of the RDF property.

To be able to understand the contents of a data source, we introduce the notion of a *signature* which is a compact representation of its contents. Using the schema directly as a signature is not the best choice, because schemas are prone to heterogeneity issues. The situation in which the same term has been used for expressing two different semantics or the situation in which the same semantics have been modelled through different terms is common. A signature should go beyond the name choices made by the data source designer and be more robust to name variations. An important feature that a signature should capture is the structure. The way data are structured in a data source is not random. It is the way the data administrator decided that the semantics of the data are best expressed. For instance, the reason that two properties are found in the same class is most likely because they model two different aspects of the same real-world concept that the class models, and they are both needed for better describing that real-world concept. This means that the signature should also capture not only what properties appear in every class, but also the co-appearance of the properties. These two principles drive the definition of the signature for classes.

Definition 1 The signature of a class $\langle c, P\rangle$ is a graph $G(V, E, f)$, such that its set of nodes is $V = \{c\} \cup N^P$, with $N^P = \{x \mid \exists\langle x, d\rangle \in P\}$, and its set of edges is $E = E^{CP} \cup E^{PP}$, with $E^{CP} = \{\langle c, n_p\rangle \mid n_p \in N^P\}$, $E^{PP} = \{\langle n_p, n'_p\rangle \mid n_p, n'_p \in N^P\}$, and with a function f being an identifier function $f|N^P \cup E^{PP} \rightarrow \mathbb{R}$.

Intuitively, a class signature is a graph that contains one node representing the class, referred to as the *class node*, and one node for every property that the class has, referred to as the *property node*. The graph has an edge between the class node and every property node, referred to as *CP* edges, standing for *class–property* edges. It also has one edge between every pair of property nodes, referred to as the *PP* edges, standing for *property–property* edges. Finally, every property node in N^P and every *PP* edge in E^{PP} are annotated with numeric value returned by the f function for that edge.

The numeric value plays a role of an identifier for a property and an identifier of the association that exists between two properties of the same class.

In what follows, for simplicity, instead of using the notation $G(V, E, f)$ for a class signature, we will use instead the equivalent more analytic form $\langle c, N^P, E^{PP}, E^{PN}, f\rangle$.

A data source is a set of classes, but these classes are not completely independent of one another. They may describe complementary information and there are mechanisms to connect them. In the relational model for instance, such mechanisms are the foreign key constraints. A similar mechanism exists also in RDF. In particular, the value of an RDF property may not be an atomic value, but a URI referencing another RDF entity. In our model, this is achieved by properties that have as a domain another class. Thus, to accommodate this important information, we consider in the signature of a data source, apart from the class signatures, a set of edges that associate a property node of one class with another class.

Definition 2 A *data source signature* is a graph $\langle C, E^{PC}\rangle$ where C is a set of class signatures, and E^{PC} is a set of edges of the form (s, e) such that $s \in N^P$ for a $\langle c, N^P, E^{PP}, E^{PN}, f\rangle \in C$, and also $e = c'$ for a $\langle c', N^{P'}, E^{PP'}, E^{PN'}, f\rangle \in C$.

Intuitively, a data source signature is a collection of class signatures with an additional set of edges between a property

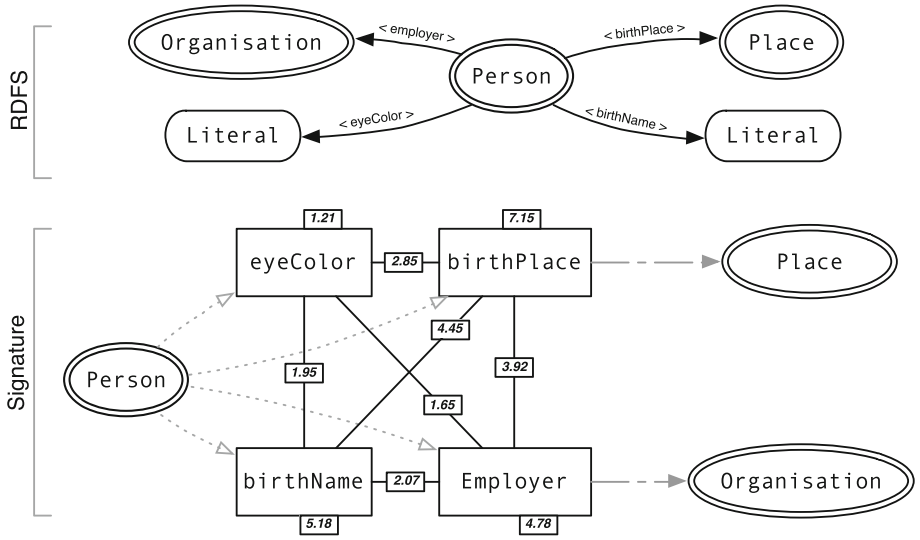

Fig. 1 A simple RDF schema and its signature

node and a class node. We refer to these edges as PC edges, which stands for *property–class edges*, and denote them as E^{PC}.

Example 1 Figure 1 illustrates a small RDF schema and its corresponding source signature. The source signature consists of three class signatures (only the one modelling *Person* is illustrated fully). The class nodes are illustrated with double oval lines and the property nodes with squared boxes. The dashed grey lines are the PC edges and the dotted the CP edges. Finally, the dotted edges are the PP edges. Note how the E^{PP} and the property nodes are annotated with the identification numbers.

2.1 Problem Statement

Our goal is to understand the topics of a data source. To do so, there is a need for some reference vocabulary in the domain of interest in which the concepts of the data source will be expressed. The reference vocabulary is a collections of classes and possible associations between them. In some sense, it can be seen as a data source. It may be provided by a domain expert explicitly or may be a reference data source.

To express the data source in the reference vocabulary, we need to express the source and the vocabulary in some common terminology, to match their contents. For this, signatures can be used.

Thus, given a reference vocabulary S_{ref} and a data source S, we need to: (1) generate the respective signatures $sig_{S_{ref}}$ and sig_S, and (2) match these two signatures to identify correspondences between their respective components, i.e. pairs of the form $\langle p, t \rangle$ such that $t \in S_{ref}$ and $p \in c$, with $c \in S$.

3 Our Solution

In dealing with the two tasks that were just mentioned (i.e. generating and matching signatures of structured data sources), there are two challenges. The first is to ensure that the signatures best represent the contents of the source (and the reference vocabulary of course), and the second is how to effectively match signatures of the data source and the reference vocabulary. Our solution for both challenges is to employ entropy and mutual information-based signatures.

The generation of the structural part of the signature, i.e. the nodes and the edges, is by definition straightforward. What is not clear is what value to assign to each node (the identifier value) and also to the PP edges.

We specialize the definition of a signature and consider as the identity value with which the properties are annotated to be the entropy of the values of the specific property. The advantage of the entropy is that it does not depend on naming choices or on structural designs. It only depends on the nature of the values that the property takes. Intuitively, our claim is that the entropy of the values of a property is the same (or highly similar) independently of the size of the data. Thus, two attributes describing, for instance, phone numbers, will have the same entropy even if the phone numbers they contain are different. A special case is the one of properties that represent referential constraints like foreign keys. Recall that these are the properties that in the signature of a data source are the origin of a PC (property–class) edge. For these properties, the computation of the entropy is not done on the values of the actual property, but on the values of the property they actually refer, i.e. the key in the class node that serves as the end point of their PC edge.

Furthermore, for the identity value with which the PP edges in the signature are annotated, we consider the mutual information of the two properties that the PP edge connects.

Example 2 In the example of Fig. 1, the computation of the identity value for the `eyeColor` is done by computing the entropy of the respective RDF property values for `eyeColor`. However, for the `birthPlace`, the entropy is not computed on the values that the `birthPlace` property has, but instead on those of the property `Place`. The identity value of the PP edge between `eyeColor` and `Employer`, on the other hand, is the mutual information between these two properties, respectively.

Unfortunately, the traditional computation method of the entropy cannot be used as it is for our purposes. It needs to be adjusted. The section that follows explains the problem and presents our alternative solution.

With the ability to generate the signatures of the data source and of the reference vocabulary, one can proceed to the matching phase. The best match that is found for each class signature is considered to be the global representation of the concept that the respective class represents, i.e. the topic. With such a common reference for every class, it is possible to realize the contents of the sources and compare them. The approach we follow is similar to the idea of a global schema provided by domain experts or of a reference ontology, both solutions that have been extensively exploited in information integration. However, in these approaches, the source schemas are manually (or with the help of a computer) matched to the global schema or ontology. In our case, we do not have any such manual matching and it is known in advance that the matching of the local signatures to those in the reference signature set is not straightforward. Since data sources very rarely provide complete information of a domain, the matching to the global signatures will be partial. However, the use of entropy which depends on the actual data values and the way the various attributes are put together to form the class structures offer a significant advantage that makes the matching successful.

In Sect. 5 we will offer an explanation on how the matching is done between the signatures of the data source and of the reference vocabulary to derive the actual matchings.

4 Likelihood Estimation of Signatures

The identifier values that annotate the nodes of the signatures and the PP edges, as previously mentioned, are computed using two basic information-theoretical quantities: the entropy and the mutual information [13].

Definition 1 (*Entropy)* Let X be a random variable representing an attribute with alphabet $\mathcal{X}$ and probability mass function $p(x|\theta), \theta \in \Theta \subseteq \mathbb{R}^p$. The entropy $H(X)$ is defined by

$$H(X) = -E_X \log p(X|\theta), \tag{1}$$

where $E(\cdot)$ denotes expectation with respect to $p(x|\theta)$.

Note that the above definition does not involve realized values for data instances, thus making the signature independent of the class represented. In particular, entropy describes the uncertainty of values in an attribute. Thus, one problem is estimation of $H(X)$ from available data instances by means of some appropriate approximation of $p(x|\theta)$. If n instances of X are available, then an estimate $\hat{\theta}$ can be obtained by some statistical estimation method, such as maximum likelihood estimation, so that $H(X)$ could be estimated using $\theta = \hat{\theta}$ in the definition above. To measure the information shared by two attributes at the time, we refer to the concept of mutual information.

Definition 2 (*Mutual information)* Let X and Y be two random variables representing attributes with alphabets $\mathcal{X}$ and $\mathcal{Y}$ with joint mass function $p(x, y|\theta^{XY})$ and marginal mass functions $p(x|\theta^X)$ and $p(y|\theta^Y)$. The mutual information of X and Y and Y is:

$$\begin{aligned} I(X;Y) &= E_{XY}\left[\log \frac{p(X,Y|\theta^{XY})}{p(X|\theta^X)p(Y|\theta^{XY})}\right] \\ &= H(X) + H(Y) - H(X,Y), \end{aligned} \tag{2}$$

where $H(X)$ and $H(Y)$ are entropies for X and Y, and $H(X, Y)$ is the entropy for the pair (X, Y).

Note that $I(\cdot;\cdot)$ measures different levels of association (or shared information) between pairs of nodes. Moreover, similarly to entropy, also the mutual information needs to be estimated from data instances. To estimate $I(X;Y)$, we need to obtain parameter estimates $\hat{\theta}^X$, $\hat{\theta}^Y$, and $\hat{\theta}^{XY}$.

The method for estimating $H(X)$, $H(Y)$ and $H(X, Y)$ from the data is crucial to obtain representative signatures. A suitable method should be able to prevent overfitting. The estimated signature does not have to perfectly replicate a specific data source, but rather provide us with a summarized representation of the concepts described in it. Overfitting is important in the presence of very large alphabets for the attributes under examination, with only a few observed instances. The elements of the alphabets with very low frequency typically inflate the overall noise, thus deteriorating the quality of the available information.

In the analysis of real data sources, we noticed that the contribution of such low-frequency elements is not negligible. Let us consider for example Fig. 2 that shows through Pareto charts the frequency distribution of some properties of the DBpedia Musical Artist class. It is evident that frequency distributions are usually right skewed, thus meaning that only few elements of the alphabets really contribute, with their high frequency, in the characterization of the entropy value for the specific property. Even if Fig. 2 includes only some properties of a selected DBpedia class, we performed

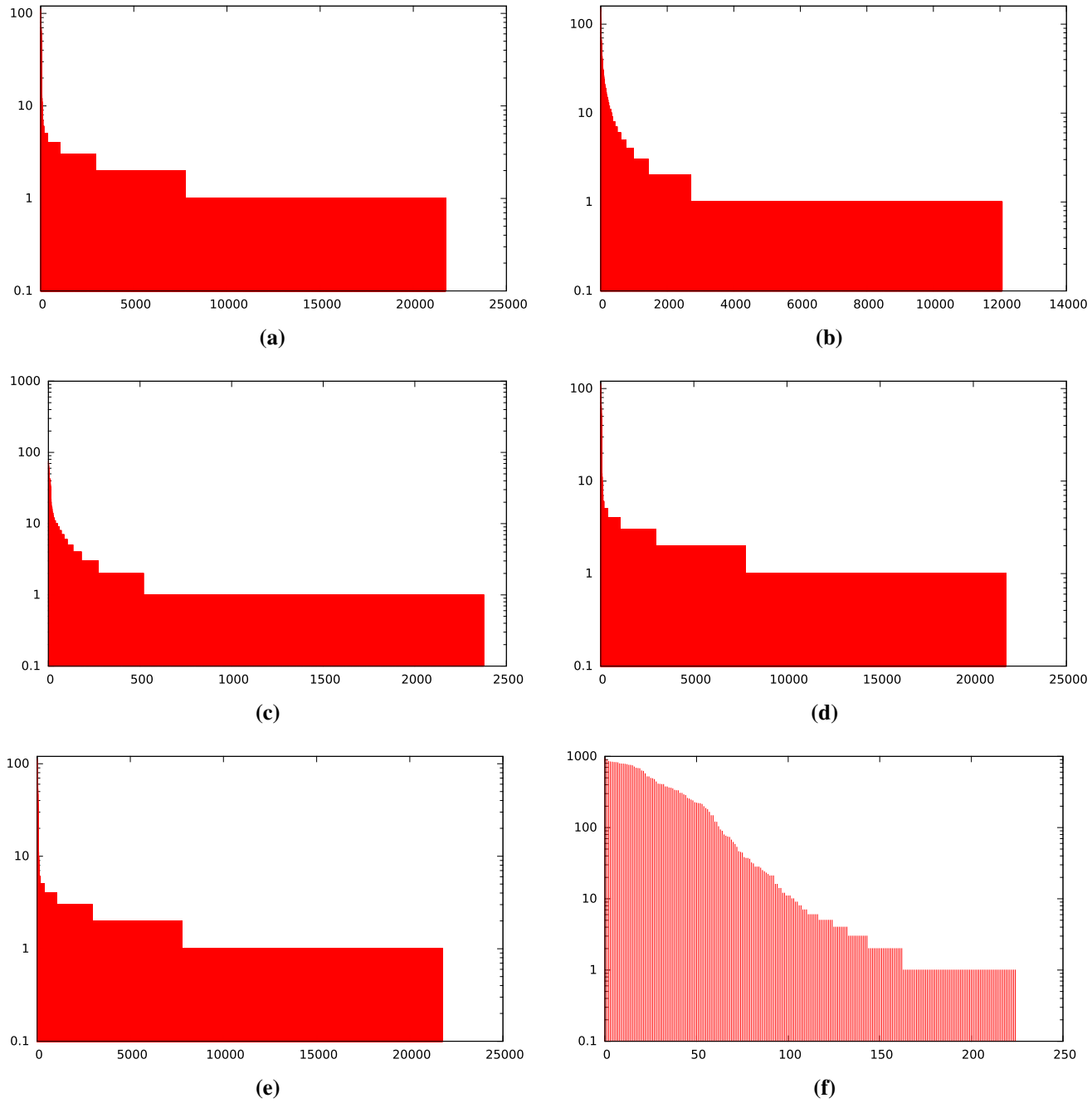

Fig. 2 Frequency distribution of some properties of the DBpedia Musical Artist class. **a** Property: birthdate. **b** Property: birthplace. **c** Property: deathplace. **d** Property: home town. **e** Property: occupation. **f** Property: Active and Start Year

an extensive analysis on the other classes and properties, obtaining similar frequency distributions.

Moreover, (1) shows that the entropy value does not have a upper bound value, since it depends on the cardinality of the attribute alphabet. As a consequence, attributes having a different number of alphabet elements in different datasets have different entropy values even when they represent the same real-world object. In Sect. 7, we will show that this problem has a big impact in real data sources, and it can affect the result accuracy. For this reason, a normalizing factor that limits the range of the entropy and mutual information values is needed.

Finally, another issue in using entropy and mutual information values is related to the high dimensionality of the problem that has a big impact on the time complexity. The high number of instances usually collected in the databases available online makes the calculation of the actual values expensive. For example, if we adopt the DBpedia Ontology

as vocabulary, the class Person (one of the 529 classes which form a subsumption hierarchy) of the DBpedia Ontology contains 832,000 instances and has 101 properties (in version 3.9). This means that the cardinality of the set E^{PP} built considering only the class Person is 5050.

4.1 Computing Entropy and Mutual Information

We implemented and tested three measures based on entropy and the mutual information: (1) a classical implementation based on Shannon's logarithmic entropy that provides us with a benchmark for the comparisons; (2) a weighted entropy and mutual information implementation that allows us to remove some "noise" provided by large regions of values with low probabilities and (3) the pseudo-additive entropy and mutual information developed by Havrda and Charvát [12]. Tsallis [21] has exploited the stability of pseudo-additive in the context of statistical mechanics. Moreover, we considered appropriate normalizations for all the above entropy measures. As a result, the range for all the measures is between 0 and 1, which allows a more fair comparison of their performance.

We begin by estimating parameters for the attributes from N available instances. In this paper, we assume a multinomial distribution for the attributes, but our approach can be easily extended to other statistical models. Suppose that a single attribute X_i, $i = 1, \ldots, k_i$ follows the multinomial model $X_i \sim \text{Mult}(\theta_i)$, where θ_i is the k_i-vector $\theta_i = (\theta_{i1}, \ldots, \theta_{ik_i})^T$, θ_{ij} corresponds to the probability of the j^{th} element of attribute alphabet, and $\sum_{j=1}^{k_i} \theta_{ij} = 1$. For instance, in RDF data sources, the alphabet of an attribute is the set of URIs and literals associated through the `rdf:range` statement of a property.

Then, given N observations on X, we have the following likelihood function

$$p(x_{i1}, \ldots, x_{ik_i}|\theta) = \frac{N!}{x_{i1}! \cdots x_{ik_i}!} \theta_{i1}^{x_{i1}} \cdots \theta_{ik_i}^{x_{ik_i}}, \tag{3}$$

where x_{ij} denotes the number of times we observe the jth element of the alphabet and $\sum_{j=1}^{i_k} x_{ij} = N$. From (3), the maximum likelihood estimates for θ_{ij} is simply the frequency $\hat{\theta}_{ij} = x_{ij}/N$, for all $j = 1, \ldots, k_i$.

Using estimated parameters, we estimate the marginal entropy for X_i for all $i = 1, \ldots, k_i$ based on the following measures:

$$\text{(i)}\ \hat{H}_i^S = -\sum_{j=1}^{k_i} \hat{\theta}_{ij} \log \hat{\theta}_{ij}, \quad i = 1, \ldots, k_i; \tag{4}$$

(Shannon entropy)

$$\text{(ii)}\ \hat{H}_i^W = -\sum_{j=1}^{k_i} \frac{\hat{\theta}_{ij}^a \log \hat{\theta}_{ij}}{\sum_{j=1}^{k_i} \hat{\theta}_{ij}^a}, \quad a > 0,\ i = 1, \ldots, k_i; \tag{5}$$

(Re-weighted entropy)

$$\text{(iii)}\ \hat{H}_i^P = -\sum_{j=1}^{k_i} \hat{\theta}_{ij} \frac{1 - \hat{\theta}_{ij}^b}{b}, \quad b > 0, \quad i = 1, \ldots, k_i. \tag{6}$$

(Pseudo-additive entropy).

The weighted entropy (5) is based on a simple variation of the classical measure, where the components are "weighted" by means of a parameter a. For the pseudo-additive entropy measure, convexity is achieved only for $b > 0$. These are empirical parameters defined, on the basis of the experiments. In the evaluated data sets, the best results were achieved using values for a and b near 1.5 and 0.5, respectively. Note that the forms in $\hat{H}_i^W$ and $\hat{H}_i^P$ reduce the noise generated by elements with low frequency and to improve the contribution by the high-frequency elements.

A normalization factor for the above measures is obtained by computing the maximum entropy. This can be achieved by replacing $\hat{\theta}_{i1}, \ldots, \hat{\theta}_{ik_i}$ with the uniform distribution $1/k_i, \ldots, 1/k_i$. Straightforward algebra gives the maximum values for (i), (ii) and (iii), respectively, $\hat{H}_{i,max}^S = \log k_i$, $\hat{H}_{i,max}^W = \log k_i$ and $\hat{H}_{i,max}^P = (1 - k_i^{-b})/b$. Finally, to normalize we simply divide each entropy measure by its maximum value.

To evaluate the relationship between two attributes, say X_i and X_l, we use the following measures of mutual information which are derived from the entropy measures above. Specifically,

$$\text{(i)}\ \hat{MI}_{il}^S = \sum_{j=1}^{k_i} \sum_{l=1}^{k_l} \hat{\theta}_{ijlm} \log \left(\frac{\hat{\theta}_{ijlm}}{\hat{\theta}_{ij} \hat{\theta}_{lm}} \right), \quad 1 \le i \le k_i, \quad 1 \le l \le k_l; \tag{7}$$

$$\text{(ii)}\ \hat{MI}_i^W = \sum_{j=1}^{k_i} \sum_{l=1}^{k_l} \frac{\hat{\theta}_{ijlm}^a}{\sum_{j=1}^{k_i} \hat{\theta}_{ijlm}} \log \left(\frac{\hat{\theta}_{ijlm}}{\hat{\theta}_{ij} \hat{\theta}_{lm}} \right), \quad a > 0, \quad 1 \le i \le k_i, \quad 1 \le l \le k_l; \tag{8}$$

$$\text{(iii)}\ \hat{MI}_i^P = -\sum_{j=1}^{k_i} \sum_{l=1}^{k_l} \hat{\theta}_{ijlm} \left[\frac{1 - \hat{\theta}_{ijlm}/(\hat{\theta}_{ij} \hat{\theta}_{lm})^b}{b} \right], \quad b > 0, \quad 1 \le i \le k_i, \quad 1 \le l \le k_l, \tag{9}$$

where $\hat{\theta}_{ij}$, $i = 1, \ldots, k_i$ and $\hat{\theta}_{lm} = 1, \ldots, k_l$ denote marginal empirical frequencies for the values of the attributes X_i and X_l, respectively, while $\hat{\theta}_{ijlm}$ represents joint empirical frequencies for the values of the attributes X_i and X_l.

4.2 Confidence Intervals to Compare the Information Computed from Two Sources

In this section, we consider the problem of comparing entropies computed from different samples. Matching based solely on point measurements is not sufficiently reliable, due to the presence of statistical errors. Thus, we propose to compare entropy measures by constructing confidence intervals for the entropy difference. Specifically, let $\widehat{H} = H(\hat{\theta}_1, \ldots, \hat{\theta}_k)$ be an arbitrary entropy method; specifically, consider entropies (i), (ii) or (iii) described in the previous section. Further, denote by $\widehat{H}_1$ and $\widehat{H}_2$ entropies on the same attribute computing, based on observations from k_1 and k_2 alphabets, respectively; $\widehat{H}_1$ and $\widehat{H}_2$ are estimated using counts in N_1 and N_2 independent samples.

Let $0 < \alpha < 1$ denote a pre-specified confidence level. A $(1-\alpha)\%$ confidence interval for the true entropy difference is

$$\mathcal{CI}(\hat{H}_1, \hat{H}_2, \alpha) = \frac{\hat{H}_1}{H_{1,max}(m_1)} - \frac{\hat{H}_2}{H_{2,max}(N_2)} \pm z_{1-\alpha/2}\sqrt{\frac{V(\hat{\theta}_1, \ldots, \hat{\theta}_{k_1}, N_1)}{H_{max}(N_1)^2} + \frac{V(\hat{\theta}_1, \ldots, \hat{\theta}_{k_2}, N_2)}{H_{max}(k_2)^2}}, \tag{10}$$

where $H_{max}(k)$ is the maximum entropy obtained by replacing the probabilities $p_1, \ldots, p_k$ with uniform probabilities $1/k, \ldots, 1/k$; so for the Shannon entropy, the maximum value is $\log m$, while for the pseudo-additive entropy we have $(1 - k^{-b})/b$. Further, in the above expression, z_q is the q-quantile for the standard normal distribution and $V_i(\cdot)$ represents an expression for an approximation of the variance of $\hat{H}_i$ obtained by the Delta method [22] . Particularly, for $j = 1, 2$, we have

$$V_j(\theta_1, \ldots, \theta_{k_j}, N_j) = \frac{1}{N_j}(\nabla \hat{H}_j)^T \begin{pmatrix} \theta_1(1-p_1) & -\theta_1\theta_2 & \cdots & -\theta_1\theta_{k_j} \\ -\theta_1\theta_2 & \theta_2(1-\theta_2) & \cdots & -\theta_2\theta_{k_j} \\ \vdots & \vdots & \ddots & \vdots \\ -\theta_1\theta_{m_j} & -\theta_2\theta_{m_j} & \cdots & \theta_j(1-\theta_{k_j}) \end{pmatrix} \nabla \hat{H}_j,$$

where $\nabla \hat{H}_j = (\partial \hat{H}_j/\partial\theta_1, \ldots, \partial \hat{H}_j/\partial\theta_{k_j})^T$ is the gradient vector of partial derivatives of the entropy function. Clearly, the form of such a vector depends on the definition of the entropy function. For Shannon's entropy, we have

$$\nabla \hat{H}_j^S = (\log\theta_1 + 1, \ldots, \log\theta_{k_j} + 1)^T,$$

while for the pseudo-additive entropy we have

$$\nabla \hat{H}_j^S = (L_a(\theta_1) + \theta_1^a, \ldots, L_a(p_{m_j}) + p_{k_j}^a)^T,$$

where the function $L_a(u) = (u^a - 1)/a, u > 0, \alpha > 0$ is the generalized logarithm.

5 Matching Signatures

The goal of the matching process is to find the signatures of the classes of the target source that match with signatures of the reference ontology. As described in Sect. 2, the signatures represent classes and properties as graphs where nodes and edges are weighted. Entropy (or pseudo-additive entropy) is used for weighting nodes and mutual information (or pseudo-additive mutual information) for the edges. Nevertheless, the effort required for computing the weights is not the same: the complexity of the mutual information computation grows quadratically with the growing of the number of the class properties, while the complexity of the entropy computation grows linearly.

Even if an accurate matching process should take into account nodes and edges, we decided to design a straightforward two-step process that requires the computation of the mutual information only when needed, thus reducing complexity in the case that the reference ontology has a high number of properties per class (i.e. avoiding comparing a huge number of possible pairs of properties). Firstly, for each property of the target source, a set of *candidate matching properties* belonging to the reference ontology is computed. The computation of matches requires taking into account the pre-computed entropy stored in GE_{index} and the confidence interval values dynamically computed on the basis of maximum entropies and entropy variances.[3]

Secondly, mutual information is computed only for those target properties that belongs to more groups, to select the best option. This selection and the computation of the final result can provide two different kinds of results: (a) `1SIG matching`, where the prototype matches the target properties into properties belonging to one single signature in the reference ontology; (b) `1+SIG matching`, where the prototype matches the target properties into properties belonging to several signatures. Obviously, `1SIG matching` is the simplest case, since it presumes that target source and reference ontology model the real world in the same way. Our technique is able to manage both the options.

Summarizing, our solution allows to perform an entropy-based match in the first place and then disambiguate the match with the support of mutual information as a second step. This can achieve a sub-optimal result (the optimal solution should consider entropy and mutual information contemporarily), but allows avoiding the $O(n^2)$, with n the

[3] For the reference ontology, the maximum entropies and entropy variances are stored in GE_{index}, while, for the target source, these measures have to be computed at runtime.

number of properties of a class, computation of mutual Information; we demonstrate in the experiment Sect. 7 that this is enough to prove the efficacy of our signature-based approach.

Example 3 Let us consider a class C_t of a target source, with five properties $(p_1, p_2, p_3, p_4, p_5)$, and a entropy-based match that returns the following candidate matching properties.

- $p_1 : \{Person_{birthYear}, Band_{startYear}\}$
- $p_2 : \{Person_{deathYear}, Band_{name}\}$
- $p_3 : \{Band_{country}\}$
- $p_4 : \{Person_{height}\}$
- $p_5 : \{\emptyset\}$

Person and *Band* are two classes of the reference ontology. In case of `1SIG matching`, the matches of properties p_1 and p_2 have to be disambiguated via the mutual information and either p_3 or p_4 are left unmatched according to the result of the previous process.

The matching process relies on the $entropy(\bar{b})$ function that takes as input a vector of discrete property values $\bar{b} = (b_1, b_2, \ldots, b_m)$ and returns its entropy value. $entropy(\bar{b})$ allows us to provide a weight to the edges E^{CP} of the signature. We can now use this representation to match the signatures of a target data source. For each data source, we build a multiset $\bar{e_i} = \{e_1, e_2, \ldots, e_l, \ldots, e_m\}$, where each element e_l represents the entropy of the l^{th} property of the class c_i, i.e. $entropy(a_l^{c_i})$. We define the *target match property set* Λe_l and *candidate class set* $c(\Lambda e_l)$ of the reference ontology for each $\bar{e_i}$

$$\Lambda e_l = \{a_j^{c_k} \mid c_k \in KB : 0 \in \mathcal{CI}(entropy(a_j^{c_k}), e_l, \alpha)\} \tag{11}$$

$$c(\Lambda_{e_l}) = \{c_k \mid \exists a^{c_k} \in \Lambda_{e_l}\}, \tag{12}$$

where e_l is the lth element of $\bar{e_i}$, and α is typically equal to 0.05.

The idea we have implemented for finding the best matches is the maximization of the *Coverage* of the matching classes belonging to the reference ontology. We define $coverage(\cdot)$ with respect to a subset of the classes $K' \subseteq K_{ref}$, where K_{ref} is the set of classes in the knowledge base is:

$$cover(e_l, K') = \begin{cases} 1, & \text{if } \exists c_k \in K' \mid a_j^{c_k} \in \Lambda e_l \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

$$coverage(\bar{e_i}, K') = \sum_{e_l \in \bar{e_i}} cover(e_l, K'). \tag{14}$$

Whenever a conflict arises on matching classes, i.e. $|c(\Lambda e_l) \cap c(\Lambda e_t)| \geq 2$, we may compute the mutual information between the properties $a_l^{c_k}$ and $a_t^{c_k}$ (corresponding to the properties having entropies matching with e_l and e_t, respectively) for all the classes in $c(\Lambda e_l) \cap c(\Lambda e_t)$. In this case, the approach proceeds greedily, trying to perform MI-based matching with properties of the classes in $c(\Lambda e_l)$ and stopping computation in case a positive match is found. The output of the matching process is a set of class $C^c \subseteq K$, ranked according to the coverage of each class after the MI-based disambiguation phase.

6 The *WHATSIT* Prototype

The *WHATSIT* prototype has been implemented in `python 2.7` and deployed on `m3.2xlarge AWS ec2` instances, with 8 vCPU and 30 GB of RAM. The functional architecture of *WHATSIT* is shown in Fig. 3. *WHATSIT* takes as an input a populated reference ontology. In our experiments, we evaluated the approach by considering DBpedia as reference ontology, and a target RDF data source. The output is the target source annotated in each property with the corresponding DBpedia property associated with the domain.

To build and compare the signatures of the sources, *WHATSIT* has to compute entropies and mutual information of properties. But, the real-time computation of mutual information for all pairs of properties is often infeasible. This is because it is common to have RDF data sources with hundreds of properties per class that would lead to a huge number of pairs. Moreover, computing mutual information for each pair of properties, often, is superfluous for the match (see Sect. 7.3). For these reasons, *WHATSIT* relies firstly on the entropy for identifying candidate matching properties, and computes the mutual information in case a disambiguation

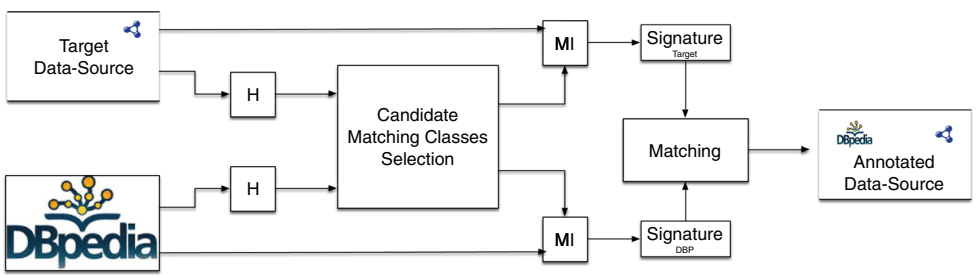

Fig. 3 The *WHATSIT* prototype functional architecture

is needed. Thus, the matching of the signatures is split into two match phases: the first, considering only the entropy; the second, considering both entropy and mutual information. Furthermore, entropies of the reference ontology can be computed in a pre-processing step and stored in an index to be checked real time. To produce candidate matching classes, the match is performed considering a confidence interval for the entropies that have been dynamically computed as described in Sect. 4.1.

7 Experimental Evaluation

The experiments proposed in this section aim to evaluate three main aspects of the approach, and, in particular, (1) the extent in which entropy-based measures are able to identify the topics described by data source properties (see Sect. 7.1); (2) the effectiveness of the signature in representing and recognizing concepts in data sources (see Sect. 7.2); (3) the effectiveness of the matching algorithm introduced in finding close signatures (see Sect. 7.3). In all the experiments, the re-weighted and pseudo additive entropies have been computed with the values of a and b equal to 1.5 and 0.5, respectively. We empirically discovered that these values typically provide good results in all the datasets considered.

The reference ontology DBpedia (version 3.9) has been adopted in our experiments as reference ontology. It conceptualizes the real world through a hierarchy structure made of 610 classes as described in the DBpedia website.[4] Each class comprises a rich set of datatype and object properties (e.g. the class Person includes more than 3k properties) and a large number of instances are provided for most of the classes (e.g. there are more than 760k instances belonging to the class Person in the English version, more than 300k belonging to the class Work). In our experiments, we considered only the DBpedia properties containing a sufficient number of instances and unique values to compute meaningful entropy values. In particular, we considered properties with at least 100 elements and assuming at least five different values. Moreover, we applied a stop-word list of terms to discard properties recording meta-information about how the class is coded in DBpedia (e.g. we did not consider properties with names containing one of the following prefixes: rdf, owl, uri, wiki, thumbnail, alias, label, etc.). In this way, we remove noise generated by "system" properties which do not convey any semantics about what the data is describing.

[4] http://wiki.dbpedia.org/Datasets/DatasetStatistics.

7.1 Experimenting Entropy as Semantic Identifier of a Property Subject

The goal of the experiments described in this section is to show that the entropy effectively identifies property topics and does not depend on the "actual" values assumed by a property in a specific data source.

First of all, our aim is to show that our implementations of the entropy-based measures are not affected by the number of instances available in the specific property. For this reason, we analyzed the entropy of samples of DBpedia properties with different dimensions. Our claim is that when the number of instances taken into account is fixed, the entropy values computed are close in all the samples.

Table 1 summarizes the results of our experiments.[5] Column 2 describes the number and the cardinality of the instances of the property shown in Column 1. The number of instances available in the properties represented in the table ranges from 4740 to 56,431. Columns from 4 to 7 show the variance of the entropy measures computed against 50 different random samples of the property with homogeneous dimensions. In particular, Column 4 shows the entropy variance of 50 random samples of a dimension equal to 10 % of the whole property. This means that, for example, the first row shows the variance of 50 random subsets, each one containing 5643 elements of the property *birthPlace* belonging to the class *Artist*. Columns 5–7 show the results obtained by the application of the same operation to 50 random subsets with dimension equal to 30, 50, and 90 % of the number of instances in the property.

The results of the experiment show that the variance is typically low, thus meaning that the entropy values are close and independent of the values randomly selected in the subsets. Moreover, the variance decreases with the increase of the number of instances taken into account. The more the instances are taken into account, the more do the entropy values converge to a fixed value.

In our second experiment, we show that entropy acts as a semantic identifier. For this reason, we compared the entropy of a property with the ones of random samples of different dimensions. Since we compare items with the same property, we expect to obtain close values. Note that entropy is sensitive to the cardinality of the property (see Sect. 4.1 and the results of the previous experiment), and the cardinality of a property containing a large number of instances is expected to be higher than the one with a small number of elements (see Fig. 2). Even if the normalization makes entropy values comparable, we cannot directly compare the entropy values

[5] Even if the table shows the analysis performed on only few properties belonging to three classes, we performed the experiment over 50+ properties belonging to 10+ classes obtaining results entirely similar to the one shown.

Table 1 Variance and mean computation of the entropy-based measures in subsets of properties with homogeneous dimensions

Element	Instances	Measure	Dimension of the subsets			
			10 %	30 %	50 %	90 %
Cl.: Artist	56431	CLA (var)	2.31×10^{-5}	7.27×10^{-6}	3.8×10^{-6}	1.64×10^{-6}
Prop.: birthPlace	c. 23667	CLA (mean)	0.6418	0.6196	0.6095	0.5991
		WEI (var)	2.14×10^{-6}	1.01×10^{-6}	6.51×10^{-7}	3.47×10^{-7}
		WEI (mean)	0.1392	0.1231	0.1175	0.1127
		PAE (var)	2.32×10^{-5}	6.45×10^{-6}	4.04×10^{-6}	1.88×10^{-6}
		PAE (mean)	0.7231	0.7248	0.7243	0.7252
Cl.: Artist	18966	CLA (var)	1.81×10^{-5}	3.3×10^{-6}	3.03×10^{-6}	9.13×10^{-7}
Prop.: nationality	c. 895	CLA (mean)	0.2165	0.2006	0.1934	0.1873
		WEI (var)	1.45×10^{-6}	1.06×10^{-6}	3.23×10^{-7}	2.48×10^{-7}
		WEI (mean)	0.0592	0.0534	0.0517	0.0499
		PAE (var)	1.82×10^{-5}	6.43×10^{-6}	4×10^{-6}	1.97×10^{-6}
		PAE (mean)	0.2567	0.2567	0.2564	0.2564
Cl.: Writer	15498	CLA (var)	8.25×10^{-5}	1.79×10^{-5}	1.36×10^{-5}	7.09×10^{-6}
Prop.: birthPlace	c. 9303	CLA (mean)	0.6824	0.6567	0.6487	0.6400
		WEI (var)	2.04×10^{-5}	7.87×10^{-6}	3.34×10^{-6}	0.06×10^{-6}
		WEI (mean)	0.1766	0.1488	0.1396	0.1324
		PAE (var)	7.45×10^{-5}	2.22×10^{-5}	2.18×10^{-5}	6.56×10^{-6}
		PAE (mean)	0.7277	0.7319	0.7329	0.734
Cl.: Writer	9455	CLA (var)	6.13×10^{-5}	1.2×10^{-5}	1.36×10^{-5}	5.02×10^{-6}
Prop.: nationality	c. 561	CLA (mean)	0.2563	0.2277	0.2168	0.2128
		WEI (var)	1.93×10^{-5}	5.58×10^{-6}	2.69×10^{-6}	1.24×10^{-6}
		WEI (mean)	0.0400	0.0261	0.0215	0.0179
		PAE (var)	7.51×10^{-5}	3.92×10^{-5}	1.73×10^{-5}	9.98×10^{-6}
		PAE (mean)	0.2160	0.2205	0.2202	0.2197
Cl.: Automobile	5121	CLA (var)	1.20×10^{-4}	5.68×10^{-5}	4.38×10^{-5}	1.68×10^{-5}
Prop.: manufacturer	c. 778	CLA (mean)	0.6584	0.6175	0.6015	0.5862
		WEI (var)	1.61×10^{-4}	6.09×10^{-5}	2.88×10^{-5}	1.62×10^{-5}
		WEI (mean)	0.2786	0.2456	0.2332	0.2249
		PAE (var)	1.44×10^{-4}	4.62×10^{-5}	3.64×10^{-5}	3.31×10^{-5}
		PAE (mean)	0.6860	0.6924	0.6929	0.6924
Cl.: Automobile	4740	CLA (var)	1.77×10^{-4}	7.43×10^{-5}	4.78×10^{-5}	2.87×10^{-5}
Prop.: transmission	c. 2403	CLA (mean)	0.6204	0.6165	0.5772	0.5652
		WEI (var)	1.48×10^{-4}	2.25×10^{-5}	1.62×10^{-5}	9.15×10^{-6}
		WEI (mean)	0.2105	0.1753	0.165	0.1544
		PAE (var)	1.92×10^{-4}	6.92×10^{-5}	5.26×10^{-5}	2.18×10^{-5}
		PAE (mean)	0.6598	0.6633	0.6630	0.6653

CLA Shannon entropy, *WEI* re-weighted entropy, *PAE* pseudo-additive entropy

of samples with the whole property population , since the measurements can be affected by error due to random sampling. To properly evaluate our results, we used a significance level of 0.05. In particular, for each property, we created 50 samples having each one dimension equal to 10, 30 and 50 % of the whole number of instances. For each "dimension", we computed the entropy for all the samples and analyzed the median value and its 95 % confidence interval. Finally, we checked if the "actual" entropy value (the one computed on all instances of the property) is contained in the confidence interval.

Table 2 Analysis of the confidence intervals: the percentages refer to the properties that correctly represent the actual entropy value

Class	# of properties	Shannon entropy			Re-weighted entropy			Pseudo-additive entropy		
		10 %	30 %	50 %	10 %	30 %	50 %	10 %	30 %	50 %
Actor	17	6	88	94	24	53	71	100	100	100
Airline	12	8	100	100	25	59	67	100	100	100
Artist	5	0	0	0	0	0	0	100	100	100
Autom.	5	20	60	80	40	40	80	100	100	100
Band	17	12	59	76	29	29	59	94	100	100
Beverage	3	33	67	100	33	67	100	100	100	100
Hockey Team	5	0	100	100	60	80	80	100	100	100
Game	4	0	100	100	25	50	100	100	100	100
Musical Artist	17	0	12	35	18	18	41	94	100	100
Painter	11	0	82	100	64	91	100	100	100	100
Politician	46	2	54	70	26	33	50	98	98	100
Rugby Club	5	0	100	100	60	60	60	100	100	100
Scientist	27	0	67	78	22	41	59	100	100	100
Soccer league	5	20	40	60	40	80	100	80	100	100
Writer	5	20	60	80	40	40	80	100	100	100

Table 2[6] shows the results of our evaluation. We observe that the Shannon entropy suffers from high bias, in particular on small datasets, since a small number of properties are within the confidence intervals (see for example the evaluation concerning 10 % of the instances where for several classes—Artist, Game, Rugby, …—no entropy value is within the confidence intervals). Conversely, the pseudo-additive accurately works well, being able to correctly approximate the actual entropy value in almost all the cases.

The comparison of DBpedia properties with their small samples guarantees that we are evaluating elements which describe the same topic (we assume that all the instances of a property describe a feature related to the specific property represented). Moreover, the large number of instances in the properties assures that we are not comparing properties with precise "copies" of them. Nevertheless, to have a more extensive evaluation, we considered 17,825 randomly selected properties, available in two snapshots of DBpedia, referring to the years 2007 and 2009 (the first containing an overall of 1.19M instances and the second 2.16M instances). It is important to note that DBPedia has evolved to such an extent that a mere 23.67 % of all property-value pairs and 48.62 % of the attribute names is common among both versions. For each kind of entropy and for each property, we computed the difference (normalized) of the values obtained in the two snapshots. Finally, we analyzed these values by calculating the mean, median and standard deviation as reported in Table 3.

All the distributions are right skewed and show a large number of values close to zero. This means that there is a large number of properties in the snapshots having similar values of entropy. Note that the pseudo-additive entropy performs better since it is able "to eliminate" more occurrences with a high difference value (see Fig. 4, where the distributions of the entropy values are shown).

Finally, we evaluated the behavior of the entropy measures on different data sources describing the same topics. For this purpose, we performed an experiment with the benchmark proposed in [14]. This benchmark is conceived for the evaluation of entity resolution approaches. It is composed of four collections, each one containing two datasets about the same domain (i.e. bibliographic and e-commerce) as shown in Table 4. The datasets describe a number of common (i.e. the same item is represented in both the sources) and different items as reported respectively in columns "Comm" and "Diff" So, for example, the first row shows that the first collection includes datasets extracted from the DBLP and ACM databases containing 2224 items which are represented in both the sources. For each attribute in the dataset, the values of Shannon entropy, re-weighted entropy, and pseudo-additive entropy have been computed. The table reports the normalized difference of the entropies for the properties common in both the datasets.

This experiment shows that the properties describing the same quality (e.g. venue, year, title, …) in different data sources have similar entropy values (the differences are close to zero in most of the cases). Moreover, in these datasets, the

[6] For the sake of simplicity, the table shows the analysis performed on only few classes. Nevertheless, we performed the experiment over 50+ classes and the results showed trends similar to the ones represented.

Table 3 Analysis of the difference of entropies computed on 17,825 properties taken from two DBpedia snapshots

	Shannon entropy	Re-weighted entropy	Pseudo-additive entropy
Mean	0.076348	0.076348	0.069418
Median	0.039265	0.029197	0.022299
Std	0.098645	0.133649	0.069418

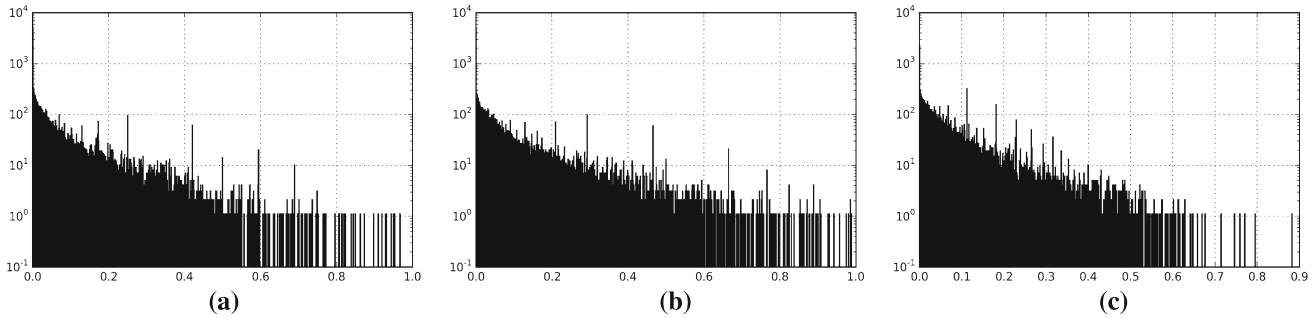

Fig. 4 Frequency distribution of the difference of entropies computed on 17825 properties taken from two DBpedia snapshots. **a** Shannon entropy. **b** Re-weighted entropy. **c** Pseudo-additive entropy

Table 4 Analysis of the difference of entropies computed on attributes of different data sources in the same domain

Domain	Sources	Comm	Diff	Shannon	Re-Weighted	Pseudo-add.
Bibliographic	DBLP	2224	463	Venue: 5.88E−02	4.20E−02	2.08E−02
	ACM			Year: 2.13E−02	2.51E−02	2.84E−03
				Title: 5.44E−03	3.40E−02	7.91E−05
				Authors: 2.48E−03	2.36E−02	5.65E−04
				Venue: 2.50	1.76	9.20E−01
Bibliographic	DBLP	5347	58,903	Year: 1.39E−01	7.01E−02	1.76E−01
	Scholar			Title: 1.02E−02	.46E−02	1.85E−02
				Authors: 1.06E−02	3.43E−02	1.99E−02
				Price: 4.63E−01	6.88E−01	1.67E−01
E-commerce	Amazon	1300	1989	Descr: 2.45E−02	7.83E−02	6.21E−03
	Google			Name.: 9.60E−03	1.98E−02	8.27E−03
				Manuf: 5.22E−02	7.68E−02	1.30E−01
				Price: 2.63E−01	3.82E−01	7.27E−02
E-commerce	Abt	1081	16	Name: 3.06E−03	5.83E−03	4.74E−04
	Buy			Descr.: 3.62E−02	5.91E−02	1.57E−02

pseudo-additive entropy performs better than the other measures, thus confirming the evaluation results achieved in the previous experiments.

7.2 Experimenting Signatures

The goal of this evaluation is to show that signatures effectively represent the data source topics. For this reason, we performed three experiments with DBpedia classes to evaluate if: (1) casual partitions of the instances related to the same class provide similar signatures; (2) the signatures of a class and the one of its superclass are close; (3) the signatures of two not related classes are different. We started the experiment by selecting three classes from DBpedia (*Writer, Artist, Automobile*) and building their signatures as shown in Fig. 5. The first signature represents a fragment of the DBpedia Writer class, including only five representative properties for simplicity. The second describes the Artist class, i.e. the superclass of Writer. Note that the classes share properties having the same name, but, since representing different entities, the values of entropy and mutual information are different. Finally, the third signature represents five properties of the Automobile class. In Fig. 5, we show the values of the pseudo-additive entropy (on the nodes) and mutual information (on the edges).

The *WHATSIT* technique relies on the specific contribution provided by entropy and mutual information alone. For this reason, we performed separate evaluations, by consider-

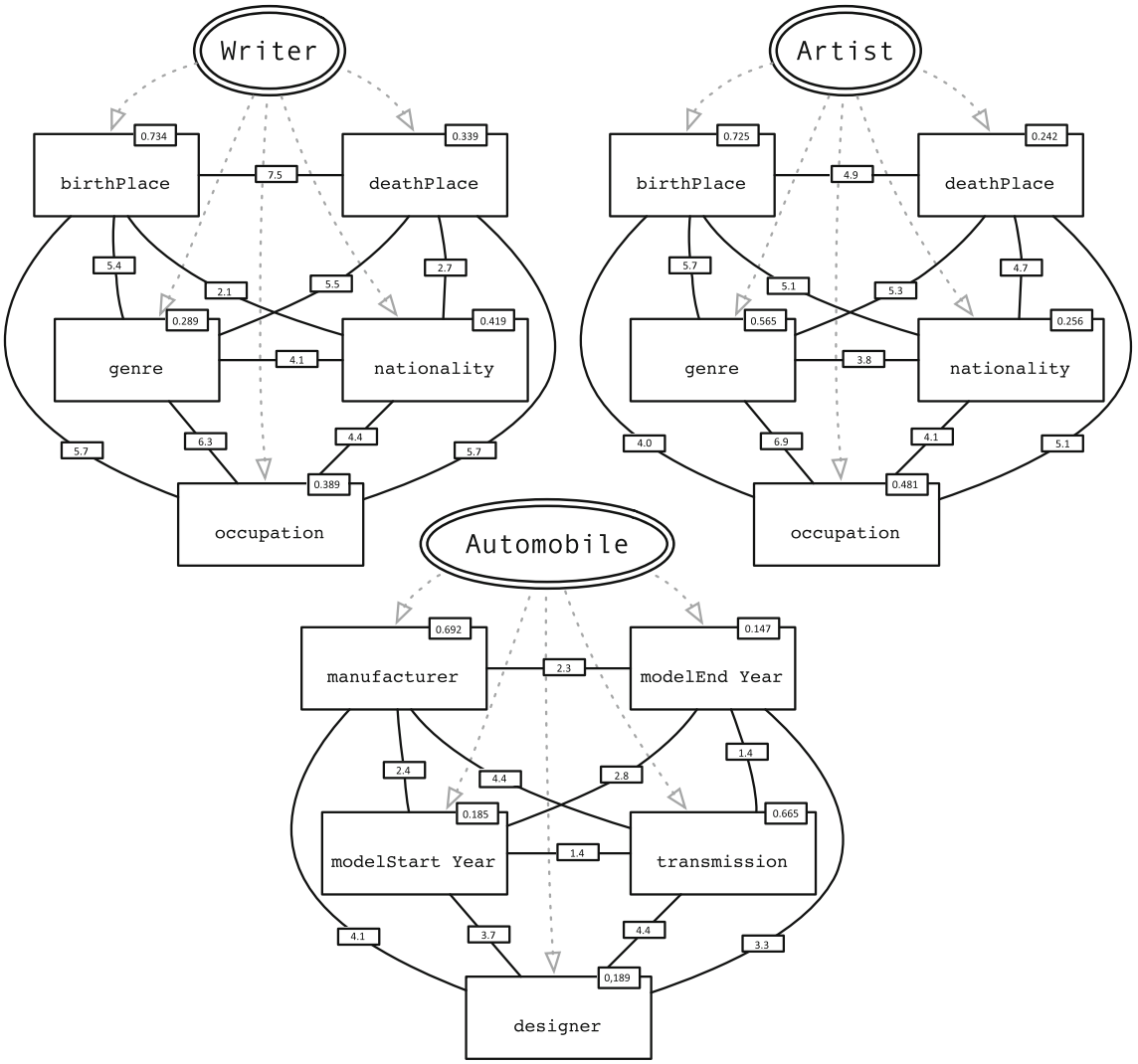

Fig. 5 The signatures of three DBpedia classes. The values in the *boxes* are pseudo-additive values for entropy and mutual information

ing firstly only the nodes (thus, measuring the contribution of the entropy) and secondly the edges (thus, measuring the contribution of the mutual information). We adopted a Euclidean distance-based metric as in [13], defined as follows. Let A and B be two equal size signatures, and a_i and b_j the entropy of the node i and j in graph A and B, respectively. Let m be an index that maps a node in graph A into the matching node in graph B (i.e. m(node in A) = matching node in B). The distance metric based on entropy for graph A and B is:

$$D = \sqrt{\sum_i (a_i - b_{m(i)})^2}.$$

An analogous distance measure can be easily defined by considering mutual information instead of entropy. The result of our experiment is shown in Table 5, where rows 1–3 compare signatures obtained by random equal-size partitions of the instances of the class Writer, Artist and Automobile (actually, the result shown is the mean of the distance measures obtained evaluating 10 random partitions). Rows 4–5 show the distances between the signature of the concept Writer and its superset Artist (with correct and random matches between

Table 5 Evaluation of the signatures

#	Comparison	Distance (H-nodes)	Distance (MI-edges)
1	Artist–Artist	0.004	0.873
2	Writer–Writer	0.007	0.142
3	Automobile–Automobile	0.004	0.349
4	Artist–Writer (best matches)	0.346	4.531
5	Artist–Writer (random matches)	0.522	5.947
6	Artist–Automobile (best matches)	0.803	8.205
7	Writer–Automobile (best matches)	0.702	8.348

Table 6 Description of the classes/properties involved in the experiment

Class	Property	# instances	Cardinality
Artist	deathPlace	16,372	285
	birthPlace	56,431	439
	genre	41,991	1409
	nationality	18,966	6202
	occupation	35,727	2437
Writer	deathPlace	6375	84
	birthPlace	15,498	157
	genre	5452	326
	nationality	9455	3796
	occupation	7585	858
Automobile	manufacturer	5121	277
	modelStart Year	1152	43
	modelEnd Year	33	0.349
	transmission	4740	250
	designer	1167	69

the properties). Rows 6–7 show the distances between the previous concepts (Writer and Artist) and the concept Automobile.

The results show that both the measures detect signatures representing similar and different concepts. As in [13], our experiment shows that the entropy alone provides a good account of the similarities between the classes. Nevertheless, the value of mutual information can support the decision about the closeness of two classes. Note that, as shown in Table 6, in this experiment we selected large and high cardinality properties. This fact lets us generalize the results observed in this fragment of DBpedia.

7.3 Matching Algorithm Evaluation

The goal of the experiments described in this section is to demonstrate the effectiveness of the matching algorithm implemented in *WHATSIT*. As a first experiment, we evaluated the algorithm with randomly selected DBpedia classes as shown in Table 7. Firstly, we selected the target classes,

Table 7 Subsets (20 %) of the instances of a DBpedia classes are considered as target classes

Target class (number of prop.)	DBpedia class	Coverage	Coverage normalized
Beverage (3)	BaseballPlayer	3/3	1
	*Beverage**	3/3	1
	BasketPlayer	2/3	0.67
Celebrity (9)	*Celebrity*	9/9	1
	Writer	7/9	0.78
	Cleri	7/9	0.78
	FootballPlayer	6/9	0.67
	Model	4/9	0.44
ChessPlayer (9)	*ChessPlayer**	8/9	0.89
	Politician	8/9	0.89
	Writer	7/9	0.78
Criminal (8)	*Criminal*	7/8	0.87
	BaseballPlayer	4/8	0.5
	Cleri	4/8	0.5
Film (5)	*Film*	5/5	1
	MusicalWork	2/5	0.4
MotorRacer (17)	*MotorRacer*	17/17	1
	Cleri	14/17	0.82
	Politician	14/17	0.82
	Actor	12/17	0.71
	Scientist	9/17	0.53
Painter (11)	*Painter*	11/11	1
	Actor	7/11	0.64
	Writer	7/11	0.64
	Airline	6/11	0.54
	BasketballPlayer	5/11	0.54

WHATSIT selected matching classes have italic font. Matching classes that require mutual information to be detected are indicated by stars

Table 8 Matching a movie dataset

Target data source			DBpedia		
Target class property	H^*_{target}	σ_{target}	Matching property:{Class}	H^*_{match}	σ_{match}
Director	0.901	0.003	Director:{Film}	0.9004	0.0012
Year	0.830	0.002	ReleaseDate:{Film}	0.8301	0.0007
Length	0.88570	0.00013	Runtime:{Film}	0.88491	0.00003
			BirthPlace:{Painter}	0.4277	0.0013
			BirthYear:{Painter}	0.8402	0.0057
			Country:{Painter}	0.1516	0.006

The star entropy H^* means normalized entropy. σ is the standard deviation

i.e. the classes that we have "to discover". To make the experiment more challenging and the dataset closer to real-world data, we considered, for each property, a sample with 20 % of the instances available in the property. Then, we considered other DBpedia classes, both in the same IS-A hierarchy and casually selected, and evaluated their matching. As shown in the table, in most of the cases, the computation of the coverage is enough to select the best DBpedia class to be associated with the input source. Only in case of tie, the computation of the mutual information is needed for the disambiguation.

Moreover, we performed a second experiment against a real movie database.[7] It originally consists of a single class with 12 properties, but, for experiment purposes, only three properties (the ones satisfying the constraints of minimum number of instances and minimal cardinality introduced at the beginning of Sect. 7), have been considered: *director, releaseDate* and. The result is a target data source containing 1406 instances. We are expecting this class to be matched with the corresponding *Film* class in DBpedia which is composed of 71629 instances.

Table 8, where the left part shows the input class properties and the right part the DBpedia corresponding properties, reports the results of this experiment. The first three rows show that *WHATSIT* finds the correct associations between the properties in the selected database and DBpedia. Note that the entropy values are close (in the confidence intervals) and relate meaningful properties. The last three rows show, as an example, the values of three properties of another randomly selected class. The entropy values are not close, thus meaning that the source is not describing the properties of a Painter.

8 Related Work

To provide users with tools for automatically understanding the content of a data source is a difficult and challenging task. The problem is well known in the IR community and commonly addressed exploiting *topic modelling* techniques [6,23] to cluster and retrieve textual documents according to their topics. These approaches are based on the assumption that the same topic can be identified in different documents by means of latent patterns in the text (i.e. relations among words), typical of every language. This assumption does not hold in the context of structured data, since the information is no longer represented as a monolithic document, but instead, as a graph, such as the *Entity-Relationship* model [7] and the *RDF* model,[8] where the relationships among concepts are explicitly modelled by means of the metadata. In the database and semantic Web literature, two main classes of solution have been proposed to automatically support the target users (e.g. data scientists, statisticians, data engineers, etc.) of structured data: *summary-based* approaches [25–28], that aim to provide a summary of a target data source; and *ontology matching* approaches [10,19], that allow to map a known ontology to the one employed to the target data source. *Summary-based* approaches aim to identify and extract a small subset of the information which is representative of the entire contents of the data source. In [25] and [26], two approaches dealing with relational databases and graphs, respectively, have been proposed. Both approaches compute the closeness between data structures and the importance of the data taking into account entropy and mutual information. In [5], the goal is to summarize an attribute domain. A mix of techniques is applied for clustering the attribute values and identifying in each cluster a single representative value. The limit of these approaches is that the produced summary maintains the same semantic of the original dataset and, therefore, a user must be able to understand such semantic (e.g. names of the classes and properties) to understand the summary itself.

Ontology-based approaches [8,10,17–19] try to match content and data structures into some reference ontology and can be generally classified, following [10], in: *schema-based* and *instance-based* mapping. The former aims to map ontologies

[7] http://perso.telecom-paristech.fr/~eagan/class/as2013/inf229/labs/datasets.

[8] http://www.w3.org/TR/WD-rdf-syntax-971002/.

relying on the schema information, e.g. trying to map classes and properties on the basis of their names, while the latter try to align ontologies using their *instances*. The intuition behind the *instance-based* approaches is that when two concepts are associated with the same set of objects (e.g. property names and their values), they are likely to be similar [18]. Thus, the *instance-based* approaches can overcome the *schema-based* approaches when it is difficult to identify the semantic similarities of the elements of the schema [13].

We note that our proposal differs from *Ontology Matching* approaches in a fundamental aspect: our goal is not to determine a fully correct (e.g. identifying class and property hierarchies) and complete match between ontologies or schemas even if the match is explicit or intentional [20]. We, instead, aim at supporting the identification of some classes of a wide reference ontology (e.g. DBpedia) that could be used to describe the topics of a data source. Our approach could be employed to support *instance-based* matching; this is an orthogonal problem that we do not tackle in this paper.

In [11], *mutual information* is employed to characterize RDFS graphs capturing the statistical association of classes and properties in an ontology; this information is then exploited to map user terms to the most appropriate element in a *schema-free* querying system. Nevertheless, in this paper we adopted a novel technique for estimating the mutual information based on likelihood. The idea of creating a data source signature starts from [13] where a dependency graph is built for supporting schema matching in a data integration approach. In this paper, we adapted the approach for RDF sources and we extended the technique with the introduction of different kinds of edges connecting nodes. Moreover, this paper radically modifies our previous proposal [3], where composite likelihood has been experimented for the same purposes. Deep evaluation showed that a best performance is achieved with the measures proposed here.

Finally, it is important to observe that Sindice.com [16], an RDF search engine, could be considered as a possible solution of the problem on hand. Nevertheless, Sindice focuses on finding triples containing particular keywords and not discovering data sources topics.

9 Conclusion and Future Work

This paper presents a proposal for providing users with an insight of data source topics. The approach relies on a reference ontology, a technique for generating signatures based on pseudo-additive versions of entropy and mutual information, and an algorithm for matching. The preliminary results evaluated, thanks to the *WHATSIT* prototype, show that the proposed measures are able to support users in identifying property domains.

Future work can be devoted to four main tasks: first, to develop and implement a graph matching algorithm able to effectively match signatures from different data sources; second, to perform an extensive evaluation of the proposed approach in different domains and with data sources taken from repositories of different nature, especially those of open data; third, to extend the technique for estimating entropy and mutual information to weighed graphs and experiment with other statistical measures for evaluating the correlation of the values to obtain more effective signatures; fourth, we are interested in combining the proposed method with other methods we have developed for understanding the meaning of keyword queries [2,4], leading to more efficient and effective query answering systems. Last but not least, since data changes over time, we are interested in understanding if and how these temporal changes affect the computed entropy values.

Acknowledgments The authors would like to acknowledge the networking support by the COST Action IC1302 (http://www.keystone-cost.eu).

References

1. Balakrishnan S, Halevy AY, Harb B, Lee H, Madhavan J, Rostamizadeh A, Shen W, Wilder K, Wu F, Yu C (2015) Applying webtables in practice. In: CIDR 2015, seventh biennial conference on innovative data systems research, Asilomar, CA, USA, January 4–7, 2015, online proceedings. www.cidrdb.org
2. Bergamaschi S, Domnori E, Guerra F, Orsini M, Trillo-Lado R, Velegrakis Y (2010) Keymantic: semantic keyword-based searching in data integration systems. PVLDB 3(2):1637–1640
3. Bergamaschi S, Ferrari D, Guerra F, Simonini G (2014) Discovering the topics of a data source: a statistical approach. In: Surfacing the Deep and the Social Web (SDSW) workshop held at international semantic web conference
4. Bergamaschi S, Guerra F, Interlandi M, Lado RT, Velegrakis Y (2016) Combining user and database perspective for solving keyword queries over relational databases. Inf Syst 55:1–19
5. Bergamaschi S, Sartori C, Guerra F, Orsini M (2007) Extracting relevant attribute values for improved search. IEEE Int Comput 11(5):26–35
6. Blei DM (2012) Probabilistic topic models. Commun ACM 55(4):77–84
7. Chen PP (1976) The entity-relationship model—toward a unified view of data. ACM Trans Database Syst 1(1):9–36
8. Choi N, Song I-Y, Han H (2006) A survey on ontology mapping. SIGMOD Rec 35(3):34–41
9. Dhar V (2013) Data science and prediction. Commun ACM 56(12):64–73
10. Euzenat J, Shvaiko P (2013) Ontology matching, 2nd edn. Springer, UK
11. Han L, Finin T, Joshi A (2012) Schema-free structured querying of dbpedia data. In: Chen XW, Lebanon G, Wang H, Zaki MJ (eds) CIKM, pp 2090–2093. ACM
12. Havrda J, Charvát F (1967) Quantification method of classification processes. Concept of structural a-entropy. Kybernetika 3(1):30–35

13. Kang J, Naughton JF (2003) On schema matching with opaque column names and data values. In: Halevy AY, Ives ZG, Doan AH (eds) SIGMOD conference, pp 205–216. ACM
14. Köpcke H, Thor A, Rahm E (2010) Evaluation of entity resolution approaches on real-world match problems. PVLDB 3(1):484–493
15. Madhavan J, Afanasiev L, Antova L, Halevy AY (2009) Harnessing the deep web: present and future. In: CIDR. www.cidrdb.org
16. Oren E, Delbru R, Catasta M, Cyganiak R, Stenzhorn H, Tummarello G (2008) Sindice.com: a document-oriented lookup index for open linked data. IJMSO 3(1):37–52
17. Rahm E (2011) Towards large-scale schema and ontology matching. In: Schema matching and mapping, pp 3–27
18. Schopman BAC, Wang S, Isaac A, Schlobach S (2012) Instance-based ontology matching by instance enrichment. J Data Semant 1(4):219–236
19. Shvaiko P, Euzenat J (2013) Ontology matching: state of the art and future challenges. IEEE Trans Knowl Data Eng 25(1):158–176
20. Srivastava D, Velegrakis Y (2007) Intensional associations between data and metadata. In: SIGMOD, pp 401–412
21. Tsallis C (1988) Possible generalization of Boltzmann–Gibbs statistics. J Stat Phys 52(1–2):479–487
22. Van der Vaart AW (2000) Asymptotic statistics. Cambridge university press, Cambridge
23. Wei X, Croft WB (2006) Lda-based document models for ad-hoc retrieval. In: SIGIR 2006: proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, Seattle, Washington, USA, August 6–11, 2006, pp 178–185
24. Wright A (2008) Searching the deep web. Commun ACM 51(10):14–15
25. Yang X, Procopiuc CM, Srivastava D (2009) Summarizing relational databases. PVLDB 2(1):634–645
26. Yang X, Procopiuc CM, Srivastava D (2011) Summary graphs for relational database schemas. PVLDB 4(11):899–910
27. Yu C, Jagadish HV (2006) Schema summarization. In: Proceedings of the 32nd international conference on very large data bases, Seoul, Korea, September 12–15, 2006, pp 319–330
28. Zhang X, Cheng G, Qu Y (2007) Ontology summarization based on rdf sentence graph. In: proceedings of the 16th international conference on world wide web, WWW 2007, Banff, Alberta, Canada, May 8–12, 2007, pp 707–716