

Matching object catalogues

Luiz André P. Leme · Daniela F. Brauner ·
Karin K. Breitman · Marco A. Casanova ·
Alexandre Gazola

Received: 14 July 2008 / Accepted: 16 September 2008
© Springer-Verlag London Limited 2008

Abstract A catalogue holds information about a set of objects, typically classified using terms taken from a given thesaurus, and described with the help of a set of attributes. Matching a pair of catalogues means to find a relationship between the terms of their thesauri and a relationship between their attributes. This paper first introduces a matching approach, based on the notion of similarity, that applies to both thesauri and attribute matching. It then describes matchings based on mutual information and introduces variations that explore certain heuristics. Finally, it discusses experimental results that evaluate the precision of the matchings and that measure the influence of the heuristics.

1 Introduction

A *database conceptual schema*, or simply a *schema*, is a high level description of how database concepts are organized. For the sake of our discussion, it suffices to assume that database

concepts are organized as classes of objects and their attributes. To *match* a *source* schema S with a *target* schema T means to find a relationship μ between the concepts in S and the concepts of T in such a way that, if two concepts s in S and t in T are related by μ , then s and t in some sense have the same meaning.

Schema matching is a fundamental issue in many database applications, such as query mediation and data warehousing [8]. The problem of query mediation becomes a challenge in the context of the Web, where the number of databases may be enormous and, moreover, the mediator does not have much control over the databases, which may join or leave the mediated environment at will.

In this context, a reasonable approach, sometimes called *extensional*, *instance-based* or *semantic*, is to detect how the same real world objects are represented in different databases and to use the information thus obtained to match the schemas. Such approach is more robust than purely syntactical approaches, but it applies only when the schemas to be matched are simple. It also depends on the ability to detect when two database objects represent the same real-world object.

A *catalogue* is a simple database that holds information about a set of objects, typically classified using terms taken from a given thesaurus. Catalogues are fairly common and can be found, for example, in e-commerce and GIS applications, such as on-line stores and gazetteers. The schema of a catalogue has a single class with a list of attributes. Matching a pair of catalogues raises three problems: (1) to match their conceptual schemas; (2) to find a relationship between the terms of their thesauri; (3) to define a way of identifying when two objects from different catalogues represent the same real-world object. Note that the last two problems are usually not considered in database schema matching.

The major contributions of this paper are threefold. First, we introduce a matching approach, based on the notion of

L. A. P. Leme (✉) · K. K. Breitman · M. A. Casanova · A. Gazola
Department of Informatics, Pontifical Catholic University of Rio
de Janeiro, Rua Marquês de S. Vicente 225, Rio de Janeiro,
RJ CEP 22451-900, Brazil
e-mail: lleme@inf.puc-rio.br

K. K. Breitman
e-mail: karin@inf.puc-rio.br

M. A. Casanova
e-mail: casanova@inf.puc-rio.br

A. Gazola
e-mail: agazola@inf.puc-rio.br

D. F. Brauner
RNP, Brazilian National Research and Education Network,
Rua Lauro Müller, 116 s. 3902, Rio de Janeiro,
RJ CEP 22290-906, Brazil
e-mail: danibrauner@rnp.br

similarity, that applies to pairs of thesauri and to pairs of lists of attributes. Second, we describe matchings based on mutual information and introduce variations that explore certain heuristics. Third, we discuss experimental results that evaluate the precision of the matchings introduced and that measure the influence of the heuristics.

This paper is organized as follows. Section 2 summarizes basic definitions. Section 3 describes a motivating example, drawn from the area of geographic information systems. Section 4 introduces our approach to catalogue matching. Section 5 describes experimental results. Section 6 presents comparisons with related work. Finally, Sect. 7 contains the conclusions and directions for future work.

2 Catalogues, catalogue queries and catalogue matching

A *thesaurus* is defined as “a structured and defined list of terms which standardizes words used for indexing” [29] or, equivalently, *the vocabulary of a controlled indexing language, formally organized so that a priori relationships between concepts (for example as “broader” and “narrower”) are made explicit* [16]. A thesaurus usually provides the following: a *preferred term*, defined as the term used consistently to represent a given concept; a *non-preferred term*, defined as the synonym or quasi-synonym of a preferred term; relationships between the terms, such as narrower term (NT), indicating that a term—the narrower term—refers to a concept which has a more specific meaning than another term—the broader term (BT).

A *catalogue* is a simple database that holds information about a set of objects. The conceptual structure of a catalogue is described by an *object type thesaurus* T and a *schema* of the form $C[A_1U_1, \dots, A_mU_m]$, where

- C is the name of the single *object class* of the schema.
- A_1 is the *object id*, which is a key of C that uniquely identifies the objects stored in the catalogue.
- A_2 is the *object type*, whose value is a term taken from T (for simplicity, we assume that the object type is unique).
- A_3, \dots, A_m is a possibly empty list of distinct attributes.
- U_i is the *domain* of attribute A_i , for each $i \in [1, m]$, which for simplicity we assume to be a subset of the *universe* U (the domain of the object type attribute is the set of terms of the thesaurus T).

We also say that T is the *thesaurus of the schema* $C[A_1U_1, \dots, A_mU_m]$. The *universe* of C is the set $U_C = U_1 \times \dots \times U_m$. An *extension* of C is an m -ary relation $C \subseteq U_C$ such that no two tuples in C have the same object id value.

A *conjunctive restriction query* over C is a conjunction of *restriction predicates*, defined over the attributes of C .

A *restriction predicate* over C is an expression of the form $A_i = v$, where A_i is an attribute of C and v denotes a value in U_i . Informal examples of conjunctive restriction queries over a catalogue of household appliances would be “select all 17 inch flat panel TVs” and “select all 220 V food processors”.

Catalogues usually provide a simple user interface that supports conjunctive restriction queries and that organizes the query results as lists of objects, which the user may browse and select the objects that catch his attention. Catalogues recently started to expose such functionality through Web services.

Let $C[A_1U_1, \dots, A_mU_m]$ and $D[B_1V_1, \dots, B_nV_n]$ be two catalogue schemas with thesauri T and W , respectively.

An *attribute matching* between C and D is a partial, many-to-many relation $\mu_A \subseteq \{A_1, \dots, A_m\} \times \{B_1, \dots, B_n\}$. We allow μ_A to be partial since some attribute of C may not match any attribute of D , and vice versa, and we let μ_A to be many-to-many to account for attributes from C that match several attributes of D , and vice-versa. We say that μ_A is *unambiguous* iff μ_A is one-to-one. Likewise, a *thesaurus matching* between C and D is a partial, many-to-many relation μ_T between terms of T and terms of W . An *instance matching* between C and D is a partial, many-to-many relation $\mu_I \subseteq U_C \times U_D$.

We say that an instance I in U_C *matches* an instance J in U_D iff $(I, J) \in \mu_I$, and likewise for attributes and thesauri terms.

A *matching* between C and D is a triple (μ_A, μ_T, μ_I) such that μ_A is an attribute matching, μ_T is a thesaurus matching and μ_I is an instance matching between C and D .

3 An informal example of catalogue matching

The area of geographic information systems (GIS) provides interesting applications of catalogues, which we explore in this section to construct an example of thesauri matching. We close the section with two brief comments on attribute and instance matching.

A gazetteer is “a geographical dictionary (as at the back of an atlas) containing a list of geographic names, together with their geographic locations and other descriptive information” [31]. For our purposes and omitting details, we consider that a gazetteer is a geographic object catalogue, where each object has as attributes:

- A unique *object ID*
- A unique *object type*, whose value is a term taken from an *object type thesaurus*
- A *name*, which takes a character string as value
- Optionally, a *location*, which approximates the position of the object on the Earth’s surface

Consistently with Sect. 2, we assume that the object type is unique. We note that geographic objects are often called *geographic features*, or simply *features* ([24]). Hence, a gazetteer thesaurus is often referred to as a *feature type thesaurus*.

Almost all gazetteers support conjunctive restriction queries using type and name restrictions, such as “select all populated places called ‘Rio de Janeiro’”, where ‘populated place’ is a term of the feature type thesaurus. Some gazetteers also allow conjunctive restriction queries that include spatial restrictions, such as “find all populated places within 10 miles of point *P*”, where *P* is defined in an appropriate coordinate (geo)reference system.

Specifically, in our example, we will use two gazetteers that are available over the Web, the GEONet Names Server and the Alexandria Digital Library Gazetteer. The GEONet Names Server (GNS) [12] provides access to the National Geospatial-Intelligence Agency (NGA) and the U.S. database of foreign geographic names, containing about 4 million features with 5.5 million names. The Alexandria Digital Library (ADL) Project [1, 14, 17] is a research program to model, prototype and evaluate digital library architectures, gazetteer applications, educational applications, and software components. The ADL gazetteer has approximately 5.9 million geographic names, classified according to the ADL Feature Type Thesaurus (FTT).

Figure 1a shows a fragment of the ADL Feature Type Thesaurus and Fig. 1b contains the equivalent fragment of the GEONet Names Server classification scheme, which strictly speaking is not a thesaurus, but just a list of terms without any thesauri relationships. Note that a simple thesauri matching strategy, based on syntactical proximity, would be of little help to match the ADL Feature Type Thesaurus and the GEONet Names Server classification scheme since the latter uses codes as thesauri terms.

In what follows, we will refer to the ADL gazetteer and the GEONet Names Server, respectively, as ADL and GNS, and to their thesauri as ADL FTT (for ADL Feature Type Thesaurus) and GNS CS (for GEONet Names Server classification scheme). We will consider only countries and cities in the examples that follow. For simplicity, we assume that the name (in English) uniquely identifies a country in both catalogues; similarly, the city name, together with the name of the upper level administrative division, uniquely identifies a city in both catalogues.

We will illustrate how to gradually construct a matching between these thesauri by post-processing the answers to queries submitted to both gazetteers. The matching may help construct a mediator to access both gazetteers or to consolidate them in a single gazetteer (as in a data warehouse application) by remapping their thesauri.

Table 1 shows sample terms collected from queries that searched the two gazetteers for the countries and cities listed in the first column. For example, if we query ADL to obtain information about “Brazil”, the answer will indicate that ADL classifies “Brazil” as “Countries”; if we then access GNS for “Brazil”, the answer shows that GNS classifies “Brazil” as “PCLI”. Therefore, we collected the first evidence that these two terms map to each other.

In fact, all five entries in Table 1 that ADL classifies as “Countries”, GNS classifies them as “PCLI”. Hence, we have better evidence that these two terms map to each other since they refer to the same five countries. If we consider that the meaning of a thesaurus term is the set of objects it classifies, then “Countries” and “PCLI” have the same meaning in this small sample. Moreover, we have not detected any conflicting classifications. The question then is how many mismatches we should allow, that is, how similar the sets of objects that two thesauri terms denote must be to consider that the two terms match.

To better explain this last remark, observe now the entries in Table 1 that ADL classifies as “Populated Places”. Note that GNS classifies three of them as “PPL” and two as “PPLA”. There are two approaches to address such situation. We may decide to match “Populated Places” with both “PPL” and “PPLA”. That is, we may decide to allow 1-to-many matchings. Alternatively, we interpret the evidence collected thus far as an indication that “Populated Places” matches “PPL” better (three entries in Table 1) than “PPLA” (two entries in Table 1).

The key questions, therefore, are what it means to “collect enough evidence”, and what to do when the mapping between terms is not one-to-one. This is addressed in Sect. 4 by introducing similarity functions that estimate how close the sets of objects that two terms denote are.

We may adopt the same strategy to match the attributes of ADL and GNS, but we would now collect information about the attribute values from the query answers. In other words, we argue that the problem of matching thesauri terms and the

Fig. 1 Fragments of the ADL and GEONET thesauri. **a** ADL FTT fragment. **b** GEONet Classification Scheme fragment

<p>Administrative Area — Populated Places — Cities — Capitals — Political Areas — Countries</p>	<table border="1"> <thead> <tr> <th>Code</th> <th>Description Text</th> </tr> </thead> <tbody> <tr> <td>PCLI</td> <td>Independent political entity ”</td> </tr> <tr> <td>AREA</td> <td>“A tract of land without homogeneous character or boundaries”</td> </tr> <tr> <td>PPL</td> <td>“Populated place”</td> </tr> <tr> <td>PPLA</td> <td>“Seat of a first-order administrative division”</td> </tr> <tr> <td>PPLC</td> <td>“Capital of a political entity”</td> </tr> <tr> <td>PCLI</td> <td>“Independent political entity”</td> </tr> </tbody> </table>	Code	Description Text	PCLI	Independent political entity ”	AREA	“A tract of land without homogeneous character or boundaries”	PPL	“Populated place”	PPLA	“Seat of a first-order administrative division”	PPLC	“Capital of a political entity”	PCLI	“Independent political entity”
Code	Description Text														
PCLI	Independent political entity ”														
AREA	“A tract of land without homogeneous character or boundaries”														
PPL	“Populated place”														
PPLA	“Seat of a first-order administrative division”														
PPLC	“Capital of a political entity”														
PCLI	“Independent political entity”														

(a)

(b)

Table 1 Results of querying countries and cities in *ADL* and *GNS*

Entry name	ADL	GNS
Brazil	Countries	PCLI
Canada	Countries	PCLI
Germany	Countries	PCLI
Italy	Countries	PCLI
Belgium	Countries	PCLI
Scotland, UK	AdministrativeArea	AREA
Wales, UK	AdministrativeArea	AREA
Rio Grande, Brazil	Populated places	PPL
Smithers, Canada	Populated places	PPL
Rio de Janeiro, Brazil	Populated places	PPLA
São Paulo, Brazil	Populated places	PPL
Cardiff, Wales	Populated places	PPLA
Asmara, Eritrea	Capitals	PPLC
Rome, Italy	Capitals	PPLC
Brussels, Belgium	Capitals	PPLC

problem of matching attributes may be both reduced to measuring set similarity: (1) similarity between the sets of objects the terms classify, in the former case; and (2) similarity between the sets of attribute values, in the latter case. Section 5 contains results about the effectiveness of this strategy.

We close with a brief comment on the problem of instance matching. In the geographic information systems domain, we have various geo-referencing schemes that associate each geographic object with a description of its location on the Earth's surface. This location acts as a universal identifier for the object, or at least an approximation thereof. In this case, we may propose that two instances match if their locations and their names are similar. This strategy works well for the geographic domain, but it depends on detecting—and matching—which attributes describe the geographic location and the name of the objects in both gazetteers. In general, one will use some form of comparing attribute values to induce instance matchings. However, the user will typically have to interfere to inform which attributes to use (such as the object location and the object name) and how to compare them.

4 Matching model

As illustrated in Sect. 3, extensional matching techniques use duplicated values to formulate hypothesis about the matching. In this section, we expand this observation and introduce the notion of similarity-induced catalogue matching.

Let $C[A_1U_1, \dots, A_mU_m]$ and $D[B_1V_1, \dots, B_nV_n]$ be two catalogue schemas with thesauri T and W , respectively. Let \mathcal{C} and \mathcal{D} be extensions of C and D .

Informally, a *similarity-based matching model* for catalogues consists of the following:

- A *similarity-based instance matching* σ_I , which is a possibly many-to-many relationship between pairs of instance representations.
- A *similarity-based attribute matching* σ_A , which is a possibly many-to-many relationship between pairs of attribute representations.
- A *similarity-based term matching* σ_T , which is a possibly many-to-many relationship between pairs of term representations.

Defining a similarity-based matching model requires addressing three problems:

- (1) Deciding on how to represent the objects—instances, attributes, thesauri terms—to be matched.
- (2) Defining a similarity measure that applies to the selected representations.
- (3) Based on the similarity measure of their representations, deciding when two objects match.

Examples of similarity measures are information content [26], mutual information [15], Dice coefficient [11], cosine coefficient [11], distance-based measurements [18], information theory-based [20] and contrast models [28]. For the sake of concreteness, we adopt as similarity measure variations of the estimated mutual information matrix, defined as follows.

Let $\mathbf{A} = (A_1, \dots, A_m)$ and $\mathbf{B} = (B_1, \dots, B_n)$ be two lists of sets. The *estimated mutual information matrix* for \mathbf{A} and \mathbf{B} is the $m \times n$ matrix *EMI* such that, for each $r \in [1, m]$ and $s \in [1, n]$:

$$EMI_{rs} = \frac{m_{rs}}{M} \log \left(M \frac{m_{rs}}{\sum_j m_{rj} \times \sum_i m_{is}} \right) \quad (1)$$

where $m_{ij} = |A_i \cap B_j|$, for $i \in [1, m]$ and $j \in [1, n]$, and $M = \sum_{i,j} m_{ij}$.

Observe that the EMI matrix is, therefore, symmetric, since $|A_i \cap B_j| = |B_j \cap A_i|$. We also say that $[m_{ij}]$ is the *co-occurrence matrix* of \mathbf{A} and \mathbf{B} .

Consider the problem of applying the estimated mutual information matrix to match thesauri terms. Assume that we have already defined an instance matching $\mu_I \subseteq \mathcal{C} \times \mathcal{D}$ for these catalogues. Recall that μ_I is a possibly many-to-many relationship between instances in \mathcal{C} and instances in \mathcal{D} . Also recall that we say that an instance I in \mathcal{C} matches and instance J in \mathcal{D} iff $(I, J) \in \mu_I$.

The *representation* of a term t of T in \mathcal{C} is the set $\mathcal{C}[t]$ of all instances in \mathcal{C} whose type is t . Likewise, the *representation* of a term u of W in \mathcal{D} is the set $\mathcal{D}[u]$ of all instances in \mathcal{D} whose type is u . This settles the first problem for thesauri terms.

To apply the concept of estimated mutual information matrix, first consider that the terms in T and W are arbitrarily

ordered as t_1, \dots, t_m and u_1, \dots, u_n , respectively. We cannot directly compute the estimated mutual information matrix for $C[T] = (C[t_1], \dots, C[t_m])$ and $D[W] = (D[u_1], \dots, D[u_n])$ since $C[t_i]$ and $D[u_j]$ are heterogeneous sets, that is, we cannot directly compute the cardinality of their intersections. However, we may redefine m_{ij} to be the cardinality of the matching set between instances in $C[t_i]$ and instances in $D[u_j]$, that is, the cardinality of $\mu_I \cap C[t_i] \times D[u_j]$. With this proviso, we may compute the estimated mutual information matrix between terms of T and terms of W , which settles the second problem for thesauri terms.

Note that, since μ_I is not necessarily one-to-one, the number of instances of $C[t_i]$ that match instances in $D[u_j]$ is not necessarily equal to the number of instances of $D[u_j]$ that match instances in $C[t_i]$. Therefore, to avoid this asymmetry, we decided to define m_{ij} to be the cardinality of $\mu_I \cap C[t_i] \times D[u_j]$.

As for the third problem, there are two directions to follow. Given the estimated mutual information matrix EMI between terms of T and terms of W , we may decide that two terms t_r and u_s match iff EMI_{rs} is the largest entry column wise and row wise, that is, we may define a thesauri matching μ_T between terms of T and terms of W as follows:

$$\begin{aligned}
 (t_r, u_s) \in \mu_T & \\
 \text{iff } EMI_{rs} \geq EMI_{rj}, & \text{ for all } j \in [1, n], \text{ with } j \neq s, \text{ and} \\
 EMI_{rs} \geq EMI_{is}, & \text{ for all } i \in [1, m], \text{ with } i \neq r \\
 \text{for each } r \in [1, m] & \text{ and } s \in [1, n] \tag{2}
 \end{aligned}$$

We say that this thesauri matching is *directly derived* from the estimated mutual information matrix. Note that Eq. (2) induces a one-to-one thesauri matching, except when there are two entries, EMI_{rs} and EMI_{vw} , such that $EMI_{rs} = EMI_{vw}$ and both satisfy Eq. (2). To force Eq. (2) to induce one-to-one matchings, we arbitrarily take the smallest r and the smallest s when there is a tie.

Alternatively, we may define that two terms t_r and u_s match iff EMI_{rs} is above a certain threshold τ_T :

$$\begin{aligned}
 (t_r, u_s) \in \mu_T & \text{ iff } EMI_{rs} \geq \tau_T \\
 \text{for each } r \in [1, m] & \text{ and } s \in [1, n] \tag{3}
 \end{aligned}$$

which induces a possibly many-to-many thesauri matching. We say that this thesauri matching is *derived* from the estimated mutual information matrix *with the help of the threshold* τ_T . This second approach requires experimentation to decide on the threshold value, but it has the advantage of accounting for potentially non one-to-one matchings.

Let us now move to the problem of applying the estimated mutual information matrix to match attributes. Deciding on how to represent attributes is a problem open to several alternative solutions.

Let A_i be an attribute of C . The *observed domain representation* of A_i in C is the set $o[C, A_i]$ such that $v \in o[C, A_i]$ iff there is an instance I in C such that the value of A_i in I is v . The observed domain representation of an attribute of D is likewise defined. We then compute the estimated mutual information matrix for the lists of sets $o[C, A] = (o[C, A_1], \dots, o[C, A_m])$ and $o[D, B] = (o[D, B_1], \dots, o[D, B_n])$, assuming that we can compare any two attribute values.

We may improve the construction of the matrix by computing m_{ij} only for pairs of attributes A_i and B_j that are of the same type (or whose type is compatible). We call this the *type compatibility heuristic*, which is obviously advantageous since it may avoid computing all $(m \times n)/2$ possible combinations of attributes from C with attributes from D (recalling that the EMI matrix is symmetric).

The next attribute representations alter in a straightforward way the computation of the estimated mutual information matrix and are introduced without repeating the details. They may also benefit from the type compatibility heuristic.

The *multiset observed domain representation* of an attribute is the multiset that contains as many elements corresponding to a single value v as the number of instances in the catalogue extension whose value for the attribute is v . Intuitively, this representation takes into account, in the similarity measure, the number of times a value occurs.

The *string domain representation* of an attribute of type *string* is the set of tokens extracted from the strings that occur as values of the attribute. This set is obtained as follows. First, tokens are extracted from a string s by splitting s in each non-word or non-numeric characters to obtain a set of substrings from s . Then, this set is reduced by eliminating substrings which are stop-words. Finally, the remaining strings are lemmatized [23]. This redefinition improves the chances of detecting that two string attributes are similar.

The *instance matching representation* of an attribute is introduced with the help of an example. Consider two book catalogues whose schemes are $B_1[ISBN_1, Type_1, Name_1, Edition_1, Rating_1]$ and $B_2[ISBN_2, Type_2, Name_2, Edition_2, Rating_2]$, with keys $ISBN_1$ with $ISBN_2$, respectively. Assume that $ISBN_1$ and $ISBN_2$ store 13-digit ISBNs, and that $Edition_1, Rating_1, Edition_2$ and $Rating_2$ store small integers (book editions typically range from 1 to 10, and book ratings from 1 to 5, say). Suppose that the correct matchings are $ISBN_1$ with $ISBN_2$, $Edition_1$ with $Edition_2$ and $Rating_1$ with $Rating_2$.

Then, the estimated mutual information matrix may correctly induce a matching between $ISBN_1$ and $ISBN_2$, since these attributes store values (ISBNs) which tend to be similar to each other (if the catalogues have a sizable number of books in common), and very different from values of the other attributes. However, the estimated mutual information

matrix may not induce the other correct matchings since the attributes involved have about the same values.

To circumvent this limitation, we define the *instance matching representation* of $Edition_1$ as the set of pairs IE_1 such that $(i, e) \in IE_1$ iff e is the edition of the book with ISBN i observed in the extension of B_1 , and likewise for the other three attributes, $Edition_2$, $Rating_1$ and $Rating_2$, generating sets IE_2 , IR_1 and IR_2 , respectively. Then, by computing the estimated mutual information matrix from such representations, we have a better chance of distinguishing the correct matchings from the incorrect matchings. Intuitively, an incorrect matching of IE_1 with IR_2 would be plausible only if a large number of books have the same edition number in B_1 as the rating value they have in B_2 , which is less likely than a large number of books occurring with the same edition number in both B_1 and B_2 (and likewise for the other pairs).

Finally, the *multiset instance matching representation* of an attribute is the variation of the instance matching representation that uses multisets, as for the multiset observed domain representation.

Given the estimated mutual information matrix EMI between attributes of C and attributes of D , we may derive an attribute matching μ_A between attributes of C and attributes of D as for thesauri terms, using Eqs. (2) or (3). However, to compute the EMI matrix, we have to decide on a representation for the attributes. We may in fact go further and (1) use several different representations, thereby generating several matrices, EMI^1, \dots, EMI^k ; (2) compute the final matrix EMI by combining EMI^1, \dots, EMI^k in a specific way, such as by taking EMI_{rs} as the maximum of $EMI_{rs}^1, \dots, EMI_{rs}^k$; (3) compute the attribute matching from the final matrix EMI , using Eqs. (2) or (3).

Finally, we briefly comment on how to match instances from C and D . We consider that a catalogue instance is *represented* by a list of some of its attribute values, that is, we admit that some attributes be left out of the instance representation.

Let $L = (a_{i_1}, \dots, a_{i_p})$ be a representation of an instance I from C , using the values of attributes A_{i_1}, \dots, A_{i_p} , and $M = (b_{j_1}, \dots, b_{j_q})$ be a representation of an instance J from D , using attributes B_{j_1}, \dots, B_{j_q} . Assume, for the sake of argument, that $p \leq q$. Assume also that we have an one-to-one attribute matching μ_A between attributes of C and attributes of D that cover all attributes in A_{i_1}, \dots, A_{i_p} . Let J' be a permutation of M , truncated up to the p th entry, such that now attribute A_{i_r} matches attribute B_{j_r} , according to μ_A , for $r \in [1, p]$. Then, adopting a strategy similar to that described above for thesauri terms and attributes, we may derive an instance matching from any vector similarity measure applied to I and J' , such as the cosine distance.

For example, in Sect. 3, we represented a geographic object by its location and name, and considered that two instances from the different gazetteers match if their geographic

location and name are similar, using cosine distance. A simpler example would be to consider that instances from different book catalogues are represented by their ISBN attributes, and that they match iff they have the same ISBN values.

Note that the instance matching defined above depends on an attribute matching and on a correct interpretation of the attributes, which may be informed by the user or, in simple cases, inferred by the system. Furthermore note that both the co-occurrence matrix for thesauri terms and the instance matching representation for attributes require that an instance matching be defined, which in turn depends on an attribute matching. In other words, such concepts are not orthogonal and require a careful engineering to avoid circularities. The examples in Sect. 5 indeed start with very simple instance matchings to derive thesauri and attribute matchings.

5 Experiments

5.1 Data sources

We conducted two experiments to assess the performance of several similarity-based matching models. The first experiment was based on data extracted from the GEOnet Names Server (GNS) and the Alexandria Digital Library gazetteer (ADL), already used in Sect. 3. The second experiment was based on data about books obtained from Amazon and Barnes & Noble. All these data sources provide Web service access, except Barnes & Noble, in which case we developed an HTML parser to capture data from query results.

For each experiment, we first defined a bootstrap set of keywords, which we used to query the databases. From the query results, we extracted the less frequent words. We then used these words to once more query the databases. This pre-processing step enhanced the probability of retrieving duplicate objects from the databases, which is essential to evaluate any extensional schema matching technique. For the first experiment, we extracted a total of 23,390 records: 3,599 from GNS and 19,791 from ADL. For the second experiment, we extracted a total of 116,201 records: 16,410 from Amazon and 99,791 from Barnes & Noble.

5.2 Experiments with gazetteers

5.2.1 Thesauri matching

The experiments described in this section focused on matching the ADL Feature Type Thesaurus (FTT) with the GEOnet Names Server classification scheme (GNS CS). Although the ADL FTT has a total of 1,262 terms, we considered only the preferred terms, which amounts to 210 terms. The GNS CS has 642 terms, organized under a single category level including nine top terms.

Table 2 A fragment of the co-occurrence matrix for ADL and GNS

GNS	ADL									
	Islands	Lakes	Mountains	Populated places	Railroad features	Reference locations	Ridges	Rivers	Streams	Waterfalls
FLLS			1	10					5	353
FRM				44		1			13	
HLL		2	177	14	1	1			5	
HLLS			136	27			2		24	
INLT				6					2	
ISL	460		1	39		3	1		18	
ISLS	20			3						
LCTY	2		7	37			2		13	
LGN		62	1	11	1				5	
LK		310	1	7	1		3		5	
LKI		2								
LKO		10								
LKS		2								
MT			74	7			3		4	
MTS			68	22	2		3		14	
PPL	32	23	83	7440	52	24	30	2	799	13
PPLA			1	6					2	
PPLL				6						
PPLX		1	1	28	2					
PS				1						
PT	3			34		181			2	
RDGE	1	1	4	21	3		101		19	1
RSTN		2	1	30	300	1	2		21	
RSTP			1	18	141		1		12	
RSV				1						
SCH				1						
SCRP			1	1					1	
SPUR		2		5			32		5	
STM	21	10	58	667	28	2	31		4732	10
STMI	2		2	90	2		2		251	

The experiments had the following characteristics:

1. Used the data extracted from ADL and GNS, as described in Sect. 5.1.
2. Adopted a simple instance matching, computed using the centroids and the names of the geographic features.
3. Tested the thesauri matching model directly derived from the estimated mutual information matrix, with each thesauri term t represented as the set of all instances whose type is t .

The instance matching adopted assumes that: (1) a geographic feature F_i is represented by the triple $(long_i, lat_i, N_i)$, where $(long_i, lat_i)$ is the centroid and N_i is the name of the

feature; (2) a matching between the attributes of ADL and GNS that store the centroid and the name of a geographic feature has been defined. These assumptions avoid the circularity problem mentioned at the end of Sect. 4, for the sake of simplicity and clarity of the experiment.

We then define that two geographical features *match* iff their centroids and their names match, computed as follows. Let F_1 and F_2 be two features. Then, we considered that their centroids *match* iff

$$\sqrt{(long_2 - long_1)^2 + (lat_2 - lat_1)^2} \leq 0.9$$

To compare N_1 and N_2 , we first computed the vector similarity v between the token vectors built from N_1 and N_2 ,

Table 3 EMI matrix corresponding to the co-occurrence matrix in Table 2

GNS	ADL								
	Islands	Lakes	Mountains	Populated places	Railroad features	Reference locations	Ridges	Streams	Waterfalls
FLLS									0.032134
HLL			0.013706						
HLLS			0.009877						
ISL	0.037034								
ISLS	0.001607								
LCTY	0.000004		0.000200	0.000208			0.000048		
LGN		0.005162							
LK		0.027295							
LKI		0.000179							
LKO		0.000893							
LKS		0.000179							
MT			0.005609				0.000075		
MTS			0.004708				0.000061		
PPL				0.108805					
PPLA			0.000028	0.000049					
PPLL				0.000107					
PPLX		0.000007		0.000412	0.000036				
PT						0.018135			
RDGE							0.009716		
RSTN					0.023947				
RSTP					0.011171				
SCRP			0.000028						
SPUR		0.000031					0.003147		
STM								0.107120	
STMI								0.004676	

Table 4 Matchings directly derived from the EMI matrix of Table 2

Matchings	
ADL	GNS
Islands	ISL
Lakes	LK
Mountains	HLL
Populated places	PPL
Railroad features	RSTN
Reference locations	PT
Ridges	RDGE
Streams	STM
Waterfalls	FLLS

different thesauri that were found to match as compared to the set of pairs of terms from the different thesauri that were defined in the reference thesauri matching. *Precision* has a similar interpretation and *fMeasure* is defined as

$$fMeasure = 2 \times precision \times recall / (recall + precision)$$

Furthermore note that the thesauri matching model directly derived from the estimated mutual information matrix is one-to-one, by definition. Therefore, the reference thesauri matching contains only one-to-one matchings.

Table 2 shows a fragment of the co-occurrence matrix and Table 3, the corresponding EMI matrix. The highlighted cells have the largest values of their respective rows and columns. Table 4 contains the thesauri matchings directly derived from the EMI matrix, according to Eq. (2). The complete analysis of the results indicates a total of 41 true positive matchings over a total of 43 correct matchings, which means a recall of 95%. By contrast, it indicates a total

taking the TF-IDF weight for each token. Then, we considered that N_1 and N_2 match iff $v \leq 0.9$.

Note that, since we are trying to obtain a thesauri matching, *recall* measures the set of pairs of terms from the

Table 5 ADL attribute list

Attribute	Description	Data type
boundingBoxX1	Longitude of the left upper corner of the bounding box containing the feature	Real
boundingBoxY1	Latitude of the upper left corner of the bounding box containing the feature	Real
boundingBoxX2	Longitude of the lower right corner of the bounding box containing the feature	Real
boundingBoxY2	Latitude of the lower right corner of the bounding box containing the feature	Real
displayName	Display name	String
footprintX	Longitude of the centroid of the bounding box of the location of the object	String
footprintY	Latitude of the centroid of the bounding box of the location of the object	String
identifier	Entry local id	String
names	Alternative names	String
placeStatus	Entry place–status (current or former)	String
relationships	Relationships with other features	String

We disregarded attributes boundingBoxX1, boundingBoxY1, boundingBoxX2, boundingBoxY2 since they actually contain the same values as footprintX, footprintY in the sample data downloaded from ADL

Table 6 GNS attribute list

Attribute	Description	Data type
adminCode1	Code for 1st administrative division	String
adminName1	Name for 1st administrative division	String
alternateNames	Alternative names	String
countryCode	Country code (ISO-3166 2-letter code)	String
countryName	Country name	String
elevation	Elevation, in meters	Real
geonameId	Identifier	String
lat	Latitude of the centroid of the bounding box of the location of the object	Real
lng	Longitude of the centroid of the bounding box of the location of the object	Real
name	Primary name	String
population	Population	Integer

Table 7 Reference attribute matchings for ADL and GNS

ADL	GNS
displayName	name
footprintX	lng
footprintY	lat
names	alternateNames

of nine *false positive* matchings, which means a precision of 88%. The *fMeasure* is then

$$fMeasure = 2 \times precision \times recall / (recall + precision) \\ = 2 \times 88 \times 95 / (95 + 88) = 91\%$$

5.2.2 Attribute matching

The experiments described in this section concentrated on matching the ADL gazetteer attribute list, shown in Table 5,

with the GNS attribute list, shown in Table 6. The experiments had the following characteristics:

1. Used the data extracted from ADL and GNS, as described in Sect. 5.1.
2. Adopted a simple instance matching, computed using the centroids and the names of the instances, as in Sect. 5.2.1.
3. Tested the family of attribute matching models directly derived from the estimated mutual information matrix, with each attribute represented as described in Sect. 4 (all models adopt the type compatibility heuristics).

Note that, since we are trying to obtain attribute matchings, *recall* measures the set of pairs of attributes from the different attribute lists that were found to match as compared to the set of pairs of attributes from the different gazetteer that were defined in the reference attribute matching (see Table 7). *Precision* has a similar interpretation and *fMeasure* is defined in terms of *recall* and *precision* as explained in Sect. 5.2.1.

Table 8 Performance of the attribute matching models directly derived from the EMI matrix

Instance matching	Observed domain	Multiset	Type compatibility	Precision (%)	Recall (%)	fMeasure (%)
<i>False</i>	<i>True</i>	<i>False</i>	<i>True</i>	71	63	67
False	True	True	True	50	50	50
True	False	False	True	50	50	50
True	False	True	True	50	50	50
True	True	False	True	50	50	50
True	True	True	True	50	50	50

The first two columns indicate the attribute representation that the model adopts (instance matching or observed domain representations)

For each line, when the value of the multiset column is *True*, it indicates that the idea of using a multiset is applied to both the instance matching and the observed domain representations

The type compatibility column is all *True*, indicating that all models use the type compatibility heuristics

The last two lines, where both the instance matching and the observed domain columns are *True*, correspond to the models based on an EMI matrix obtained by taking, for each entry, the maximum value from the EMI matrix computed using the instance matching representation and the EMI matrix computed using the domain value representation

Table 9 A fragment of the co-occurrence matrix for attributes of ADL and GEONames

GNS	ADL				
	displayname	footprintx	footprinty	names	relationships
admincode1	23	9	15	25	11
adminname1	156			110	160
alternatenames	252		1	371	59
countrycode	4			4	4
countryname	59			21	62
elevation	4	2	1	4	
lat	8	222	1250	8	
lng	2	1323	445	2	
name	381			382	43
population					1

Table 10 EMI matrix corresponding to the co-occurrence matrix in Table 9

GNS	ADL				
	displayname	footprintx	footprinty	names	relationships
admincode1	0.00086	0.00008	0.00025	0.00095	0.00046
adminname1	0.00666			0.00387	0.00983
alternatenames	0.01079			0.01832	0.00197
countrycode	0.00016			0.00016	0.00024
countryname	0.00266			0.00052	0.00399
elevation	0.00017	0.00004	0.00000	0.00017	
lat		0.00331	0.05750		
lng		0.06029	0.01023		
name	0.01811			0.01787	0.00104
population					0.00008

Table 8 shows the performance results for the attribute matching models directly derived from the estimated mutual information matrices computed using different attribute

representations and combinations thereof. For the first model in Table 8, we show in Tables 9, 10 and 11 the corresponding co-occurrence matrix, the estimated mutual information

Table 11 Attribute matchings corresponding to the third model in Table 8

ADL	GNS
footprintX	lng
footprintY	lat
names	alternateNames
relationships	adminName1

Table 12 Amazon attribute list

Attribute name	Description	Data type
author		String
edition		Integer
index	Book classification	String
isbn		String
label		String
listPrice		Real
productGroup		String
productType		String
publisher		String
title		String
url		URL

matrix and the directly derived attribute matchings. The complete analysis of the results for the first model indicates a total of three *true positive* matchings over the total of four correct alignments, which means a *recall of 75%* of the total correct matchings. By contrast, it indicates 1 *false positive* matching, which means a *precision of 75%*. The *fMeasure* is then

$$fMeasure = 2 \times precision \times recall / (recall + precision) = 75\%$$

5.3 Experiments with Book Catalogues

The experiments described in this section repeat the experiments of Sect. 5.2.2 for the Amazon and the Barnes & Noble book catalogues. Tables 12 and 13 show the Amazon and the Barnes & Noble attribute lists, whereas Table 14 contains the reference attribute matchings.

In this experiment, we assumed that: (1) an instance from ADL is represented by the values of attributes *title*, *author*, *publisher* and *isbn*; (2) an instance from Barnes & Noble is represented by the values of attributes *name*, *by*, *publ* and *isbn-13*; (3) the attributes in these two lists match (see however the observation about *isbn-13* below). We considered that two instances *match* iff their representations are similar, using as similarity measure the cosine distance with TF-IDF, and a threshold of 0.9.

Table 15 shows performance results for the attribute matching models directly derived from the estimated mutual information matrix, and it should be interpreted as Table 8. Table 15 indicates that the matching models based on the

Table 13 Barnes & Noble attribute list

Attribute name	Description	Data type
by	Author	String
category	Book classification	String
isbn-13	The 13-digit International Standard Book Number	Integer
name	Title of the book	String
numberOfPages	Number of pages	Integer
pubDate	Publication date	Date
publ	Publisher	String
salesRank	Number of times that other titles sold more than this book title	Integer
subject		String

Table 14 Reference attribute matchings for the Amazon and Barnes & Noble book catalogues

	Amazon	Barnes & Noble
author		by
index		category
publisher		publ
title		name

instance matching representation for attributes (lines 3 to 6) do not have the best performance. This can be explained in part since, in this sample data, the number of instances from both catalogues that match is fairly low. For the first model in Table 15, Tables 16 and 17 show the occurrence and the estimated mutual information matrices computed, and Table 18 shows the attribute matchings derived.

An interesting observation can be made regarding ISBN values. Starting in 2007, the 13-digit ISBN began to replace the 10-digit ISBN. The Amazon book catalogue stores both numbers, with the attribute *isbn* holding the old 10-digit ISBN and the attribute *ean* (not used in the experiment), the new 13-digit ISBN. The Barnes & Noble book catalogue stores only the new 13-digit ISBN (the attribute *isbn-13*). Differently from a syntactical approach, which would wrongly match *isbn* with *isbn-13*, due to their syntactical similarity, our instance-based technique did not match *isbn* with *isbn-13*, since obviously these attributes have no common values (they are in fact omitted from Tables 16 and 17).

The date attributes also never matched due to differences in format. Indeed, Amazon stores dates in the format “YYYY-MM-DD”, while the Barnes & Noble stores the publication date as “Month, YEAR”. To solve this problem, we would have to consider a more sophisticated strategy to compare dates.

6 Related work

Rahm and Bernstein [25] deliver an early survey of schema matching techniques. Euzenat and Shvaiko [10] provide an

Table 15 Performance results for the attribute matching models directly derived from the EMI matrix

Instance matching	Observed domain	Multiset	Type compatibility	Precision (%)	Recall (%)	fMeasure (%)
False	True	False	True	100	100	100
False	True	True	True	100	75	86
True	False	False	True	57	100	73
True	False	True	True	57	100	73
True	True	False	True	57	100	73
True	True	True	True	60	75	67

Table 16 A fragment of the co-occurrence matrix for attributes of Amazon and Barnes & Noble corresponding to the first model of Table 15

Amazon	Barnes & Noble							
	by	category	name	numberOfPages	pubDate	publ	salesRank	subject
author	2580	1	927	3	3	377	2	26
edition	60	1	137	23	22	49	29	2
index	1	1	1			1		1
label	3		62	148	12	1	166	
listprice	5	1	12			6		4
productGroup	913	1	1138	12	20	890	10	48
productType	1642	1	3785	149	77	761	159	62
publisher			3				1	
title	2580	1	927	3	3	377	2	26
url	60	1	137	23	22	49	29	2

Table 17 EMI matrix corresponding to the co-occurrence matrix in Table 16

Amazon	Barnes & Noble							
	by	category	name	numberOfPages	pubDate	publ	salesRank	subject
author	<i>0.018669</i>							
edition	0.000422			0.002287	0.001848		0.000885	
index		<i>0.014983</i>				0.001488		0.012317
listPrice				0.005492	0.000726		0.004622	
productGroup		0.014797				0.001495		0.012277
productType	0.000001							0.000001
Publisher	0.005598					<i>0.014014</i>		
title			<i>0.015650</i>	0.000947	0.000751			
url			0.029205				0.024320	

account of ontology matching techniques. Following their classification, the techniques described in this paper is extensional and based on data analysis and statistics. Bernstein and Melnik [2] list the requirements for model management systems that support schema mappings, to which the work reported in this paper contributes.

Bilke and Naumann [3] describe an extensional technique based on similarity algorithms. Brauner et al. [5] adopt the same idea to match two thesauri. Wang et al. [30] describe

a technique based on query probing to match Web databases which relies on human intervention to select a set of typical instances used in the probing. Brauner et al. [7] apply this idea to match geographical database Web services. Brauner et al. [6] describe a matching algorithm based on measuring the similarity between the attribute domains of distinct Web databases. Madhavan et al. [22] propose the use of a set of schemas and mappings to help the schema matching algorithms. The authors use predictor algorithms that mea-

Table 18 Attribute matchings corresponding to the first model in Table 15

Amazon	Barnes & Noble
author	by
index	category
publisher	publ
title	name

sure the similarity between schema elements, adopted in the PayGo architecture [21].

Contrasting with Wang et al. [30] and Brauner et al. [7] we avoid the use of a global schema and a set of global instances, which are sometimes hard to define. Section 5 also explore how the precision of the attribute matchings is influenced by the attribute representations adopted.

Castano et al. [9] describe the H-Match algorithm to dynamically match ontologies. H-Match provides, for each concept from an ontology, a ranked list of similar concepts in the other ontology. Four matching models are used to dynamically adjust the matching process to different levels of richness of the ontology descriptions. Spertus et al. [27] evaluate the performance of six similarity measures, used to recommend online communities to members of related communities from of the Orkut social network and adopt the L2 vector normalization (L2-Norm) measure.

7 Conclusions and future work

In this paper, we proposed an approach to match pairs of catalogues. The approach is classified as extensional since it uses instances stored in the catalogues, and is based on the notion of similarity. To provide the foundations of the discussion, we first defined the concepts of thesauri, attribute and instance matchings, and discussed how to use similarity functions to induce matchings. Specifically, we adopted the estimated mutual information (EMI) matrix to measure similarity and defined how to derive thesauri and attribute matchings from the EMI matrix. We also called attention to the fact that attributes may have alternative representations, which impact the computation of the EMI matrix. Finally, we illustrated the approach with experiments using data from catalogues available on the Web. The experiments also measured the influence of the alternative attribute representations on the performance of the attribute matchings derived.

The results described in the paper admit at least three extensions, as described in Casanova [19]. First, although we concentrated on just two catalogues, we may extend the overall approach to match multiple catalogues by computing the EMI matrix between any two catalogues. Second, in addition to one-to-one matchings, we may derive many-to-many matchings by using the EMI matrix as in Eq. (3), as well as by

adopting other similarity functions. The results are still promising, but they require a training step to calibrate the threshold value (see Eq. (3)), and additional parameters, when other similarity functions are adopted (see [19]). Finally, we have not discussed how to gradually construct the matchings as new data from the catalogues are available, which is typical of a query mediation environment. We refer the reader to Brauner et al. [4,6] for discussions about this issue.

Acknowledgments This work is partly supported by CNPq under grants 142103/2007-1, 301497/2006-0, 550930/2007-8 and 140417/2005-2.

References

- ADL (1999) Alexandria digital library gazetteer. Map and Imagery Lab, Davidson Library, University of California, Santa Barbara, CA. Copyright UC Regents. <http://www.alexandria.ucsb.edu/gazetteer>
- Bernstein P, Melnik S (2007) Model management 2.0: manipulating richer mappings. In: Proc. 2007 ACM SIGMOD Intl. Conf. on Management of Data, pp 1–12. ACM Press, New York, NY, USA
- Bilke A, Naumann F (2005) Schema matching using duplicates. In: Naumann F (ed) Proc. 21st Int'l. Conf. on Data Engineering, pp 69–80
- Brauner DF, Casanova MA, Milidiú RL (2006) Mediation as recommendation: an approach to design mediators for object catalogues. In: OTM Confederated International Workshops and Posters. Montpellier, France, 29 October–3 November 2006. Lecture Notes in Computer Science, vol 4278, pp 46–47. ISSN 0302-9743
- Brauner DF, Casanova MA, Milidiú RL (2007a) Towards gazetteer integration through an instance-based thesauri mapping approach. In: Advances in geoinformatics. Springer, Heidelberg, pp 235–245
- Brauner DF, Gazola A, Casanova MA (2008) Adaptive matching of database web services export schemas. In: Proc. Int'l. Conf. on Enterprise Information Systems, Barcelona, Spain
- Brauner DF, Intrator C, Freitas JC, Casanova MA (2007b) An instance-based approach for matching export schemas of geographical database web services. In: Vinhas L, da Rocha Costa AC, (eds) IX Proc. Brazilian Symposium on Geoinformatics, pp 109–120
- Casanova MA, Breitman KK, Brauner DF, Marins AL (2007) Database conceptual schema matching. *Computer*, IEEE Computer Society, pp 102–104
- Castano S, Ferrara A, Montanelli S, Racca G (2004) Semantic information interoperability in open networked systems. In: Proc. Int'l. Conf. on Semantics of a Networked World (ICSNW), in cooperation with ACM SIGMOD 2004, Paris, France
- Euzenat J, Shvaiko P (2007) Ontology matching. Springer, New York
- Frakes W, Baeza-Yates R (1992) Information retrieval: data structure and algorithms. Prentice Hall, Englewood Cliffs, NJ, USA
- GNIS (2005) Geographic Names Information System, U.S. Department of the Interior, U.S. Geological Survey, Reston, USA. <http://geonames.usgs.gov/>
- GNS (2006) GEOnet Names Server, U.S. National Geospatial-Intelligence Agency, USA. <http://gnswww.nga.mil/geonames/GNS>
- Hill L, Frew J, Zheng Q (1999) Geographic names: the implementation of a gazetteer in a geo-referenced digital library. In: D-Lib. <http://www.dlib.org/dlib/january99/hill/01hill.html>

15. Hindle D (1990) Noun classification from predicate-argument structures. In: Proc. 28th annual meeting of the association for computational linguistics, pp 268–275, Morristown, NJ, USA
16. ISO-2788 (1986) Documentation—guidelines for the development of monolingual thesauri, International Standard ISO-2788, 2nd edn, pp 11–15
17. Janée G (2004) ADL Gazetteer Service Protocol v.1.2. <http://www.alexandria.ucsb.edu/gazetteer/protocol/>
18. Lee J (1993) Information retrieval based on conceptual distance in Is-A hierarchies. *J Document* 49(2):188–207
19. Leme LAP, Casanova MA (2008) Schema matching using similarity models. Technical Report 28/08. Department of Informatics, PUC-Rio
20. Lin D (1998) An information-theoretic definition of similarity. In: Proc. 15th Int'l. Conf. on Machine Learning, pp 296–304, Madison, WI
21. Madhavan J, Cohen S, Dong XL, Halevy AY, Jeffery SR, Ko D, Yu C (2007) Web-scale data integration: you can afford to pay as you go. In: CIDR, pp 342–350. <http://www.crdrrdb.org>
22. Madhavan J, Madhavan J, Bernstein P, Doan A, Halevy A (2005) Corpus-based schema matching. In: Bernstein P (ed) Proc. 21st Int'l. Conf. on Data Engineering ICDE 2005, pp 57–68
23. Manning CD, Schütze H (2000) Foundations of statistical natural language processing, chap 8, pp 265–271. The MIT Press, Cambridge, England
24. Percivall G (2003) OpenGIS® Reference Model, Document number OGC 03-040, Version 0.1.3, Open GIS Consortium, Inc
25. Rahm E, Bernstein P (2001) A survey of approaches to automatic schema matching. *VLDB J* 10(4):334–350
26. Resnik P (1995) Using information content to evaluate semantic similarity in a taxonomy. In: Proc. 14th Int'l. Joint Conf. on Artificial Intelligence, pp 448–453
27. Spertus E, Sahami M, Buyukkokten O (2005) Evaluating similarity measures: a large-scale study in the orkut social network. In: Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery and data mining, Chicago, IL, USA, August 21–24, pp 678–684
28. Tversky A (1977) Features of similarity. *Psychol Rev* 84(4):327–352
29. UNESCO (1995) UNESCO Thesaurus. United Nations Educational, Scientific and Cultural Organization. <http://www.ulcc.ac.uk/unesco>
30. Wang J, Wen J, Lochovsky F, Ma W (2004) Instance-based schema matching for web databases by domain-specific query probing. In: Nascimento MA, Özsu MT, Kossmann D, Miller RJ, Blakeley JA, Schiefer KB (eds) Proc. 13th Int'l. Conf. on Very Large Data Bases, pp 408–419, Toronto, Canada
31. Wordnet (2005) Wordnet—a lexical database for the English language. Cognitive Science Laboratory, Princeton University, Princeton, NJ, USA. <http://wordnet.princeton.edu>