

An Instance-based Approach for Matching Export Schemas of Geographical Database Web Services

Daniela F. Brauner, Chantal Intrator, João Carlos Freitas, Marco A. Casanova

Departamento de Informática – PUC-Rio
Rua Marquês de São Vicente, 225 – 22.453-900 – Rio de Janeiro – RJ – Brazil
{dani, cintrator, jcsfreitas, casanova}@inf.puc-rio.br

***Abstract.** This paper describes a semantic approach for matching export schemas of geographical database Web services, based on the use of a small set of typical instances. The paper also contains an extensive experiment, in the context of two gazetteers, Geonames and the ADL gazetteer, to illustrate the approach.*

1. Introduction

A *database Web service* consists of a Web service interface with operations that provide access to a backend database. When a client sends a query to a database Web service, the backend engine submits the query to the backend database, collects the results and delivers them to the client. The *export schema* describes the subset of the backend database schema that the database Web service makes visible to the clients [Sheth and Larson, 1990]. Usually, the export schema consists of a flat table, which does not have complex dependencies with other elements of the backend database schema. In addition, a Web service typically announces its interfaces using Web Service Definition language – WSDL, a W3C standard.

The goal of this paper is to present a semantic approach for matching export schemas of geographical database Web services, based on the use of a small set of typical instances. The paper illustrates the approach with an extensive experiment that uses two gazetteers, Geonames and the ADL gazetteer, an ISO-complaint, pre-defined geographical global schema, and a set of typical geographical locations.

The paper is organized as follows. Section 2 discusses related work. Section 3 introduces the proposed semantic schema matching approach. Section 4 describes the experiment and discusses open issues. Finally, Section 5 contains the final considerations and suggestions for future work.

2. Related work

According to Rahm and Bernstein (2001), schema matching is a basic problem in many database application domains, such as Web-oriented data integration. The *match* operation takes two schemas as input and produces a mapping between elements of the two schemas that correspond to each other. Many techniques for schema and ontology matching have been proposed to automate the match operation. Rahm and Bernstein (2001) present a survey on several schema matching approaches.

Hess et al. (2006) proposes G-Match, an algorithm for geographic ontology matching. G-Match takes two different geographic ontologies as input, measures the similarities of their concepts by considering class and attribute names (string similarity), and hierarchical and topological relationships, producing as output a list of similarity measures between the concepts from the two ontologies. For class name and attribute name matching, they use WordNet [Wordnet, 2006] to feed the algorithm with synonyms. This approach therefore assumes that syntactical and structural similarity implies semantic proximity, which is often not warranted. Natural language dictionaries may be useful, perhaps even multi-language dictionaries (e.g., English-Japanese) to deal with schemas using terms in different languages. In addition, domain- or enterprise-specific dictionaries may sometimes be essential to deal with organizational standards, such as abbreviations for schema element names.

Wang et al. (2004), propose a unified solution to the problem of database schema matching. Their approach is based on an instance-based schema matching technique by domain-specific query probing, applied to Web databases. A Web database is a backend database available on the Web and accessible through a query interface. In particular, a Web database has two different schemas, the interface schema (**IS**) and the result schema (**RS**). The interface schema of an individual Web database consists of data attributes over which users can query, while the result schema consists of data attributes that organizes the query results that users receive.

This approach is based on three observations about Web databases:

1. Improper queries often cause search failure, that is, return no results. For the authors, improperness means that the query keywords submitted to a particular interface schema element are not applicable values of the database attribute to which the element is associated. For instance, if you submit a string to query an attribute that is originally defined as an integer, you get an error. As an example, if you submit a *latitude* value to the search element *place name*.
2. The keywords of proper queries that return results very likely reappear in the returned result pages.
3. There is a global schema (**GS**) for Web databases of the same domain [He and Chang, 2003]. The global schema consists of the representative attributes of the data objects in a specific domain.

The query probing technique consists of exhaustively sending keyword queries to the query interface of different Web databases, and collecting their results for further analysis. Based on the third observation, they assume, for a specific domain, the existence of a pre-defined global schema, and a number of sample data objects under the global schema, called *global instances*. For Web databases, they deal with two kinds of schema matching: *intra-site schema matching* (that is, matching global with interface schemas, global with result schemas, and interface with result schemas) and *inter-site schema matching* (that is, matching two interface schemas or two result schemas).

The data analysis is based on the second observation. Given a proper query, the results will probably contain the re-occurrence of the submitted value (referring to the values of the attributes of the global instances). The results will be organized using the HTML sent to Web browser. Thus, the re-occurrence of the query keywords in the

returned results can be used as an indicator of which query submission is appropriate (i.e., to discover associated elements in the interface schema). In addition, the position of the submitted query keywords in the result pages can be used to identify the associated attributes in the result schema.

The query probing process is based on the following workflow. Given a Web database with its query interface, an element identification component first locates qualified input elements. Then, a query submission component exhaustively submits the attribute values of the global instances into those identified input elements. After collecting the returned results for all submitted queries, a wrapper induction component induces a regular-expression wrapper composed of HTML-tags. Next, a data extraction component employs the induced wrapper to extract structured data objects from query result pages and arrange them into a data table. Finally, the re-occurrences of submitted queries in the columns of this table are counted and stored into a query occurrence cube. Then, using a projection function, say sum, the 3-dimensional cube is projected onto three Query Occurrence Matrices (front, top and left), which exactly reflect the relationship between pairs of the three schemas (i.e., GS and IS, IS and RS, and GS and RS). The main research issue now becomes how to find the correspondence between a pair of schemas in the projection matrices. In this context, to discover *intra-site schema matching* they applied the concept of mutual information. Moreover, to discover *inter-site schema matching*, they applied the idea of vector similarity used in the Vector Space Model from information retrieval [Salton, 1989].

In our paper, we will focus only on the query probing process applied to match export schemas (as result schema in [Wang et al., 2004]), as explained on the next section.

3. Instance-based Schema Matching

Based on the query probing process of Wang et al. (2004), we propose an instance-based approach for schema matching, in the context of geographical database Web services.

A database Web service is a well-specified service that provides Web access to a database. By well-specified, we mean that the service has a XML document (preferably, but not necessarily, a WSDL document) that describes the input attributes (interface schema) and the output attributes (export schema). Note that, by using an XML description, we do not require the definition of an HTML wrapper to locate qualified input (query interface attributes) and output elements (attributes of the result set).

Our first prototype of the schema matching process (Figure 1) starts with the XML descriptions of a set of database Web service, a previously defined global schema, and a set of global instances. For each global instance, the query formulator component creates queries based on the global instances and the Web service input attributes. The query submission component is responsible for submitting the queries to the Web service engine. After collecting the returned results for all submitted queries and storing them in local tables, the result analyzer component analyzes the global instances and the result set looking for re-occurred values, and creates the occurrence matrix.

The occurrence matrix is created with the number of re-occurrences of the global instance value in the result set. For each re-occurred value, the re-occurrence is

attributed to the correspondent export schema attribute (occurrence matrix rows) and the correspondent global schema attribute (occurrence matrix columns). An individual cell corresponds to the re-occurrence frequency of matching the global schema attribute with the export schema attribute.

Given an occurrence matrix, we define that an attribute of the export schema matches an attribute of the global schema as follows. We first normalize the matrix elements (the re-occurrence values) by dividing them by the overall number of returned entries. Then, we define that a pair of attributes *match* iff the normalized value is greater than a given threshold, namely, 0,2 (that is, 20%) in this case, based on our experiments observation.

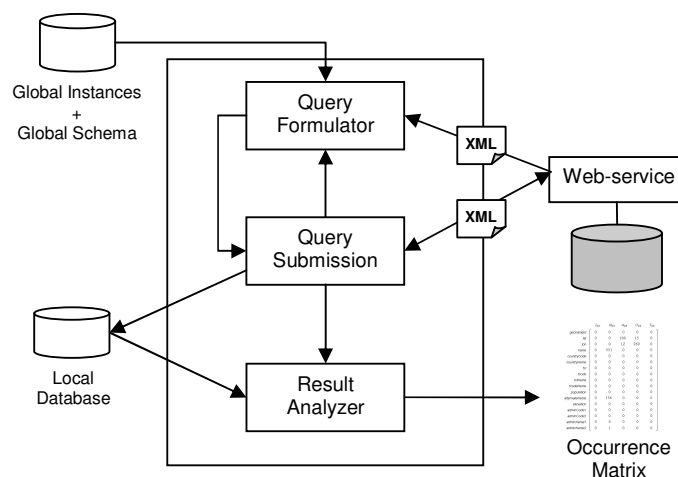


Figure 1. Instance-based Schema Matching Process

4. Experimental Approach

4.1. Global Schema and Global Instances

We designed a set of experiments using two gazetteers, available as database Web services. The experiments adopt a global schema capturing the essential characteristics of a gazetteer, and depend on a set of global instances, describing popular geographic place.

The global schema (see Figure 2) is based on the ISO 19112:2003, the recommended model for spatial referencing using geographic identifiers [ISO/TC211-ISO19112, 2003]. In detail, the global schema contains two classes, GeoInstance and GeoType, based on the ISO recommended classes, SI_LocationInstance and SI_Location Type, respectively. Table 1 and Table 2 show the attributes of classes GeoInstance and GeoType.

The global instances represent the data that will be submitted as queries to the Web services. The global instance set contains a set of geographic place names carefully chosen to cover a number of representative geographic locations. Firstly, we manually

compile a list of 36 popular geographic names that would form the basic reference database. Then, we submitted these 36 distinct names to the Geonames.org Web service. As expected, each of the name-queries returned several results, and we ended-up with thousands of entries for merely 36 initial names. The cleaning-up process of the instances was accomplished by taking the response of each query and manually locating the “most famous” place. All entries, except the “most famous” places, were discarded. The remaining entries were stored in a local database, following the global schema specified on Table 1 and Table 2. As an example, Table 3 shows a fragment of the global instances set.

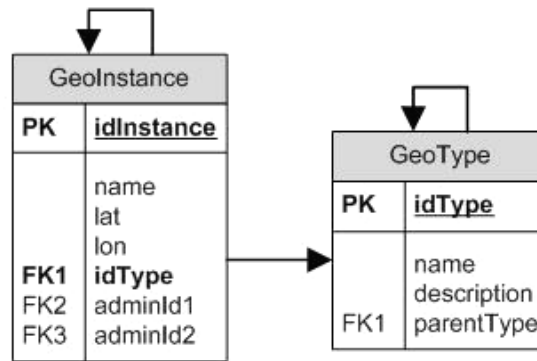


Figure 2. E-R Model of the proposed Geographical Global Schema

Table 1. Attributes of the *GeoInstance* Global Schema element

Attribute name	Description	Data Type
idInstance (I_{GS})	The entry identifier	Integer
name (N_{GS})	The entry name	String
lat (A_{GS})	The entry latitude	Double
lon (O_{GS})	The entry longitude	Double
idType (T_{GS})	GeoType code - Foreign Key (FK) for GeoType.idType	Integer
adminId1 ($A1_{GS}$)	First-order division - FK for GeoInstance.idInstance	Integer
adminId2 ($A2_{GS}$)	Second-order division - FK for GeoInstance.idInstance	Integer

Table 2. Attributes of the *GeoType* Global Schema element

Attribute name	Description	Data Type
idType	The entry identifier	Integer
name	The entry name	String
description	The entry description	String
parentType	The entry parent (broader term) - FK for GeoType.idType	Integer

Table 3. Global Instances fragment

idInstance	name	lat	lon	idType	adminId1	adminId2
175	Galapagos Islands	0.0	-90.5	4	73	-
52	Alps	46.4166667	10.0	15	165	-
149	Atlantic Ocean	10.0	-25.0	9	-	-
90	Niagara Falls	43.083416155	-79.06627052	21	123	-
16	Pão de Açúcar	-22.9472	-43.1561	14	101	-
34	Mississippi River	29.1510582	-89.2533842	19	109	-

4.2. Experimental Geographical Databases Web Services

The set of experiments uses two gazetteers, available as database Web services, Geonames¹ and the Alexandria Digital Library (ADL) Gazetteer². In our experiments, we accessed both gazetteers through their *search-by-place-name* Web services.

Geonames is a gazetteer that contains over six million features categorized into one of nine classes and further subcategorized into one out of 645 feature codes. Geonames was created using data from the National Geospatial Intelligence Agency (NGA) and the U.S Geological Survey Geographic Names Information System (GNIS). Geonames services are available through the Web services. Table 4 presents the Geonames export schema. Figure 3 shows a fragment of the XML response of this service.

The ADL Gazetteer comprises both US and non-US geographic place names. The ADL Gazetteer, and can be accessed through XML- and HTTP-based requests [Janée and Hill, 2004]. Table 5 presents the ADL export schema. Figure 4 shows a fragment of the XML response of this service.

4.3. Experimental Results

Our experiments were executed using the instance-based schema matching process described in Section 3. We used the set of global instances (Section 4.1) and the Web services provided by the ADL Gazetteer and the Geonames (Section 4.2). From the 36 global instances submitted to the gazetteers, the ADL Gazetteer returned 459 registries and the Geonames, 703 registries.

The re-occurrence detection method was created as follows: for the *name* attributes, we used the standard string comparison operator to detect the occurrence of a string in another. For the *latitude* and *longitude* attributes, we first truncated the value to four digits before comparing the values.

¹ Geonames - <http://www.geonames.org>

² ADL Gazetteer - <http://www.alexandria.ucsb.edu/gazetteer>

Table 4. Geonames Search Web Service Export Schema

Attribute name	Description	Data Type
geonameId	The entry identifier	String
name	The entry primary name	String
alternateNames	Comprises the set of alternative names	String
countryCode	The entry country code (ISO-3166 2-letter code)	String
countryName	The entry country name	String
population	The population of the instance	Number
lat	The entry latitude	Number
lng	The entry longitude	Number
fcl	The feature type super class code	String
fclName	The feature type super class name	String
fcode	The feature type classification code	String
fcodeName	The feature type classification name	String
elevation	The entry elevation, in meters	Number
admCode1	Code for first administrative division	String
admName1	Name for first administrative division	String
admCode2	Code for second administrative division	String
admName2	Name for second administrative division	String
timezone	Timezone description	String

Table 5. ADL Gazetteer Search Web Service Export Schema

Attribute name	Description	Data Type
identifier	The entry identifier	String
placeStatus	The entry place-status (current or former)	String
name	The entry primary name	String
displayName	The entry primary name as it is displayed	String
footprintX	The entry longitude	Number
footprintY	The entry latitude	Number
class	The entry class	String
thesaurus	The thesaurus of the entry class	String
names	Comprises the set of alternative names	names
relationships	The entry "partOf" relationships	String

```

<?xml version="1.0" encoding="UTF-8" ?>
- <geonames style="FULL">
  <totalResultsCount>1</totalResultsCount>
  - <geoname>
    <name>Amazon River</name>
    <lat>-0.1666667</lat>
    <lng>-49.0</lng>
    <geonameId>3407729</geonameId>
    <countryCode>BR</countryCode>
    <countryName>Brazil</countryName>
    <fcl>H</fcl>
    <fcode>STM</fcode>
    <fclName>stream, lake, ...</fclName>
    <fcodeName>stream</fcodeName>
    <population />
    <alternateNames>Orellana,Rio Amazonas,Rio Marañon,Rio Solimões,Rio
    Solimões,Rio el Amazonas,Rio Amazonas,Rio Marañón,Rio el
    Amazonas,Salimoes River,Solimoes</alternateNames>
    <elevation />
    <adminCode1>00</adminCode1>
    <adminName1 />
    <adminCode2 />
    <adminName2 />
    <timezone dstOffset="-3.0" gmtoffset="-3.0">America/Belem</timezone>
  </geoname>
</geonames>

```

Figure 3. XML response fragment of Geonames.org Search Web Service

```

<?xml version="1.0" encoding="UTF-8" ?>
- <gazetteer-service xmlns="http://www.alexandria.ucsb.edu/gazetteer"
  xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.alexandria.ucsb.edu/gazetteer
  http://www.alexandria.ucsb.edu/gazetteer/protocol/gazetteer-service.xsd" version="1.2">
- <query-response>
  - <standard-reports>
    - <gazetteer-standard-report>
      <identifier>adlgaz-1-1410143-3a</identifier>
      <place-status>current</place-status>
      <display-name>Amazon River - Brazil</display-name>
      - <names>
        <name primary="true" status="current">Amazon River</name>
        <name primary="false" status="current">Solimoens</name>
        <name primary="false" status="current">Salimoes River</name>
        <name primary="false" status="current">Orellana</name>
        <name primary="false" status="current">Marañon, Rio</name>
        <name primary="false" status="current">Amazonas, Rio</name>
        <name primary="false" status="current">Amazonas, Rio el</name>
        <name primary="false" status="current">Solimoes, Rio</name>
      </names>
      - <bounding-box>
        - <gml:coord>
          <gml:X>-49.0</gml:X>
          <gml:Y>-0.1667</gml:Y>
        </gml:coord>
        - <gml:coord>
          <gml:X>-49.0</gml:X>
          <gml:Y>-0.1667</gml:Y>
        </gml:coord>
      </bounding-box>
      - <footprints>
        - <footprint primary="true">
          - <gml:Point>
            - <gml:coord>
              <gml:X>-49.0</gml:X>
              <gml:Y>-0.1667</gml:Y>
            </gml:coord>
          </gml:Point>
        </footprint>
      </footprints>
      - <classes>
        <class thesaurus="ADL Feature Type Thesaurus" primary="true">streams</class>
        <class thesaurus="NIMA Feature Designation" primary="false">STM (stream)</class>
      </classes>
      - <relationships>
        <relationship relation="part of" target-name="UTM grid GE28" />
        <relationship relation="part of" target-name="JOG Sheet Number SA22-0" />
        <relationship relation="part of" target-name="Brazil" target-identifier="adlgaz-1-19-19" />
      </relationships>
    </gazetteer-standard-report>
  </standard-reports>
</query-response>
</gazetteer-service>

```

Figure 4. XML response fragment of ADL Gazetteer Search Web Service

As a result, we obtain two occurrence matrices (Figure 5). Figure 5 (a) and (b) show, respectively, the occurrence matrix between the global schema and the Geonames export schema, and the occurrence matrix between the global schema and the ADL Gazetteer export schema. As an example, Figure 5 shows that *name* from Geonames had 551 re-occurrences of the values of the attribute N_{GS} from the global schema (N_{GS} represents the attribute *name* of the global schema, see Table 1). For instance, when a global instance *name* value (N_{GS}) as “Mount Everest” was submitted to the Geonames search Web service, the value “Mount Everest” reappeared six times as the value of the attribute *name* from Geonames (Table 6). The final re-occurrence value between the attribute *name* from Geonames and the attribute N_{GS} from the global schema is the sum of the reoccurrence of all 36 names of the submitted global instances to the Geonames service.

	I_{GS}	N_{GS}	A_{GS}	O_{GS}	T_{GS}	AI_{GS}	$A2_{GS}$
<i>geonameId</i>	0	0	0	0	0	0	0
<i>lat</i>	0	0	188	15	0	0	0
<i>lon</i>	0	0	12	269	0	0	0
<i>name</i>	0	551	0	0	0	0	0
<i>countryCode</i>	0	0	0	0	0	0	0
<i>countryName</i>	0	0	0	0	0	0	0
<i>fcl</i>	0	0	0	0	0	0	0
<i>fcode</i>	0	0	0	0	0	0	0
<i>fclName</i>	0	0	0	0	0	0	0
<i>fcodeName</i>	0	0	0	0	0	0	0
<i>population</i>	0	0	0	0	0	0	0
<i>alternateName</i>	0	156	0	0	0	0	0
<i>elevation</i>	0	0	0	0	0	0	0
<i>adminCode1</i>	0	0	0	0	0	0	0
<i>adminCode2</i>	0	0	0	0	0	0	0
<i>adminName1</i>	0	8	0	0	0	0	0
<i>adminName2</i>	0	1	0	0	0	0	0

(a)

	I_{GS}	N_{GS}	A_{GS}	O_{GS}	T_{GS}	AI_{GS}	$A2_{GS}$
<i>identifier</i>	0	0	0	0	0	0	0
<i>placeStatus</i>	0	0	0	0	0	0	0
<i>name</i>	0	459	0	0	0	0	0
<i>displayName</i>	0	352	0	0	0	0	0
<i>footprintX</i>	0	0	14	134	0	0	0
<i>footprintY</i>	0	0	94	12	0	0	0
<i>class</i>	0	0	0	0	0	0	0
<i>thesaurus</i>	0	0	0	0	0	0	0
<i>names</i>	0	435	0	0	0	0	0
<i>relationships</i>	0	24	0	0	0	0	0

(b)

Figure 5. Occurrences matrices between (a) Geonames.org Export Schema and GS, and (b) ADL Gazetteer Export Schema and GS

Table 6. Reoccurrence of “Mount Everest” in a fragment of the results of Geonames.org Search Web Service

<i>geonameId</i>	<i>lat</i>	<i>lng</i>	<i>name</i>	<i>country Code</i>	<i>fcode</i>
1283416	27.9833	86.9333	Mount Everest	NP	MT
1004850	-28.15	29.16667	Mount Everest	ZA	MT
4122419	33.78733	-93.3804	Mount Everest Church	US	CH
4334114	29.94326	-90.0904	Mount Everest Baptist Church	US	CH
4341122	29.94104	-90.089	Second Mount Everest Baptist Church	US	CH
4694788	32.70374	-96.7881	Greater Mount Everest Baptist Church	US	CH

Given an occurrence matrix, we define that an attribute of the export schema matches an attribute of the global schema iff the normalized value is greater than 0,2 (as explained in Section 3).

For instance, Figure 5 (a) shows that *name* and *alternateName* from Geonames matches with N_{GS} from the global schema (N_{GS} represents the attribute *name* of the global schema; see Table 1). More precisely, the attribute N_{GS} had 551 reoccurred values on the attribute *name* of the Geonames export schema, what means approximately 78% of the overall of 703 entries returned by the Geonames service. The attribute *alternateName* had 156 reoccurred values, what means approximately 22%. The attributes *lat* and *lon* from Geonames correctly match with A_{GS} and O_{GS} from the global schema, respectively, with approximately 27% and 38%. By contrast, the attribute O_{GS} had 15 re-occurred values on the attribute *lat* from Geonames, which means approximately 2% of the overall reoccurred values. This value indicates that O_{GS} does not match *lat*.

Using the same procedure for the ADL gazetteer, the occurrence matrix in Figure 5 (b) shows that attributes *name*, *displayName* and *names* from ADL all align with N_{GS} from the global schema, with approximately 100%, 77% and 95%, respectively, relative to a total of 459 returned entries. Other correct matches are *footprintX* and *footprintY* from ADL with O_{GS} and A_{GS} from the global schema, respectively.

4.4. Further considerations on global instances

In our experiments, we observed some important issues that need further consideration.

First, the design of the global schema obviously influences the matching process. In our experiments, we observed that some attributes of the export schemas have no direct correspondence with any of the attributes of the global schema, such as the attribute *population* of the Geonames export schema. To overcome this problem, we suggest that the global schema be extended automatically. The idea is to add to the global schema, on demand, new attributes found on export schemas. When a new attribute appears in an export schema, the system must add this new attribute to the global schema and populate the global instances set with its values. The new global schema attribute should be labeled as “recommended” and, after it receives a sufficiently large number of recommendations (evidences coming from other export schemas), it becomes an “active” attribute. However, this issue brings new challenges to this approach: update the old global instances with the correct values of the new attribute; and, define the threshold value for the number of recommendations above which the recommended attribute becomes active.

Another issue related to the design of the global schema refers to attributes with temporal aspects. For example, suppose that the global instance set holds data from 2007, but a specific Web service provides data from 1970. In this case, the values of attribute *population*, say, would never re-occur on the returned data.

Second, as already observed in [Wang et al., 2004], the performance of the instance-based matching approach depends on the selection of the global instances. We must carefully select the global instance set in such way that:

1. global instances are representative of the overall application domain to maximize the chance that the global instances are indeed found in the database Web services to be considered;
2. global instances have attribute values that do not match with too many attribute values of an export schema.

Consider again the geographic names domain. Then, to achieve (1), the global instance set must cover, as much as possible, the variety of types of geographic features, and it must contain “famous” places (w.r.t. the region considered) .

Condition (2) is a difficult point, however. For example, if data about the country “Brazil” as a global instance, then “Brazil” will occur several times as *countryName* of several instances returned from the Geonames service. Indeed, an attribute that indicates an administrative area should not be analyzed alone. Instead, it must be analyzed in conjunction with other attributes to eliminate the risk of matching a global instance name that occurs as an administrative name of other global instances. If we have an expressive number of administrative areas as global instances, we will probably generate false matchings between the global attribute *name* and other attributes of the export schema. This problem indeed generalizes to geographic features used as aggregates of other geographic features, such as a mountain range.

As a second example where Condition (2) fails, in our experiments, we noticed that city, state and country names frequently occur inside the character string that defines a geographic feature name. This is the case, for example, with the values of the attribute *displayName* of the ADL Gazetteer, which is used to store the place name as it must to be displayed in the interface of an ADL Gazetteer client. For example, the display name of “Niagara Falls” is “Niagara Falls – Niagara County – New York – United States”.

Finally, errors in the attribute values (or in the interpretation of the attribute values) generate another issue that may create false matchings. For instance, in Geonames, we noticed that "Niagara Falls" occurs as an alternate name for a hotel named "Glengate Hotel", located in the state of “Ontario” in “Canada”, and that "American Canyon" occurs as an alternate name for a hotel called "Gaia Napa Valley Hotel", located in the state of "California" in the “United States”.

5. Conclusion

In this paper, we proposed a semantic approach, using instances, for matching export schemas of geographical database available through Web services. We also described experiments using two real Web gazetteers services. Based on the experiments, we listed some important issues that must be considered when designing the global schema and when selecting the global instances set.

As future work, we intend to improve the instance-based schema matching process in several directions. We plan to improve the re-occurrence detection method; execute a validation step to define formally a threshold to the proportion between reoccurrence values; and prototype a Web databases services mediator as a proof of concept. In addition, we intend to analyze how to improve the performance of the method by including, for instance, the automatic updating of the global schema.

References

- He, B. and Chang, C. C. (2003), Statistical schema matching across Web query interfaces. In: Proc. ACM SIGMOD Conference.
- Hess, G., Iochpe, C. and Castano, S. (2006), An Algorithm and Implementation for GeoOntologies Integration. GEOINFO 2006, Campos do Jordão, Brazil.
- ISO/TC211 (2003). "ISO 19112:2003 Geographic information — Spatial referencing by geographic identifiers". International Standard 19112. Technical report of Technical Committee ISO/TC 211.
- Janée, G. and Hill, L. L. (2004), ADL Gazetteer Protocol. Alexandria Digital Library Project. Available at <http://www.alexandria.ucsb.edu/gazetteer/protocol/>
- Rahm, E. and Bernstein, P. A. (2001), A Survey of Approaches to Automatic Schema Matching, The VDLB Journal, vol. 10, pp. 334–350.
- Salton, G. (1989), Automatic text processing: the transformation, analysis, and retrieval of information by computer, Addison-Wesley, Reading, MA.
- Sheth, A. and Larson, J. (1990), Federated database systems for managing distributed, heterogeneous and autonomous databases. ACM Computing Surveys, v.22 (3), set. 1990. pp.183-236. Available at: <http://portal.acm.org/citation.cfm?id=96602.96604>
- Wang, J., Wen, J. Lochovsky, F.H. and Ma, W. (2004). Instance-based schema matching for web databases by domain-specific query probing, In Proceedings of 30th Intl. Conference on Very Large Data Bases, pp. 408-419, 2004.
- Wordnet (2006), Wordnet 3.0 - a lexical database for the English language. Cognitive Science Laboratory, Princeton University, Princeton, NJ – USA. Available at: <http://wordnet.princeton.edu>