

W-Ray: A Strategy to Publish Deep Web Geographic Data

Helena Piccinini^{1,2}, Melissa Lemos¹, Marco A. Casanova¹, Antonio L. Furtado¹

¹Department of Informatics – PUC-Rio – Rio de Janeiro, RJ – Brazil
{hpiccinini, melissa, casanova, furtado}@inf.puc-rio.br

²Diretoria de Informática – IBGE – Rio de Janeiro, RJ – Brazil
helena.piccinini@ibge.gov.br

Abstract. This paper introduces an approach to address the problem of accessing conventional and geographic data from the Deep Web. The approach relies on describing the relevant data through well-structured sentences, and on publishing the sentences as Web pages, following the W3C and the Google recommendations. For conventional data, the sentences are generated with the help of database views. For vector data, the topological relationships between the objects represented are first generated, and then sentences are synthesized to describe the objects and their topological relationships. Lastly, for raster data, the geographic objects overlapping the bounding box of the data are first identified with the help of a gazetteer, and then sentences describing such objects are synthesized. The Web pages thus generated are easily indexed by traditional search engines, but they also facilitated the task of more sophisticated engines that support semantic search based on natural language features.

Keywords: Deep Web, Geographic Data, Natural Language Processing.

1 Introduction

Unlike the *Surface Web* of static pages, the *Deep Web* [1] comprises data stored in databases, dynamic pages, scripted pages and multimedia data, among other types of objects. Estimates suggest that the size of the Deep Web greatly exceeds that of the Surface Web – with nearly 92,000 terabytes of data on the Deep Web versus only 167 terabytes on the Surface Web, as of 2003. In particular, Deep Web databases are typically under-represented in search engines due to the technical challenges of locating, accessing, and indexing the databases. Indeed, since Deep Web data is not available as static Web pages, traditional search engines cannot discover data stored in the databases through the traversal of hyperlinks, but rather they have to interact with (potentially) complex query interfaces.

Two basic approaches to access Deep Web data have been proposed. The first approach, called *surfacing*, or *Deep Web Crawl* [16], tries to automatically fill HTML forms to query the databases. Queries are executed offline and the results are translated to static Web pages, which are then indexed [15]. The second approach, called

federated search, or *virtual integration* [4, 18], suggests using domain-specific mediators to facilitate access to the databases. Hybrid strategies, which extend the previous approaches, have also been proposed [21].

Despite recent progress, accessing Deep Web data is still a challenge, for two basic reasons [20]. First, there is the question of scalability. Since the Deep Web is orders of magnitude larger than the Surface Web [1], it may not be feasible to completely index the Deep Web. Second, databases typically offer interfaces designed for human users, which complicates the development of software agents to interact with them.

This paper proposes a different approach, which we call *W-Ray* by analogy with medical X-Ray technology, to published conventional and geographic data, in vector or raster format, stored in the Deep Web. The basic idea consists of creating a set of natural language sentences, with a simple structure, to describe Deep Web data, and publishing the sentences as static Web pages, which are then indexed as usual. The use of natural language sentences is interesting for three reasons. First, they lead to Web pages that are acceptable to Web crawlers that consider words randomly distributed in a page as an attempt to manipulate page rank. Second, they facilitate the task of more sophisticated engines that support semantic search based on natural language features [5, 24]. Lastly, the descriptions thus generated are minimally acceptable to human users. The Web pages are generated following the W3C guidelines [3] and the recommendations published by Google to optimize Web site indexing [9].

This paper is organized as follows. Section 2 describes how to publish conventional data. Section 3 discusses how to describe geographic data in vector format. Section 4 extends the discussion to geographic data in raster format. Finally, Section 5 contains the conclusions. The details of the *W-Ray* approach can be found in [22].

2 The *W-Ray* approach for conventional databases

2.1 Motivation and overview of the approach

The *W-Ray* approach to publishing conventional data as Web pages proceeds in two stages. In the first stage, the designer manually defines a set of database views that capture which data should be published, and specifies templates that indicate how sentences should be generated. The second stage is automatic and consists of materializing the views, translating the materialized data to natural language sentences, with the help of the templates, and publishing the sentences as static Web pages.

Note that metadata, typically associated with geographic data, can be likewise processed.

As an alternative to synthesizing natural language sentences, one might simply format the materialized view data as HTML tables. However, this is not a reasonable strategy for at least two reasons. First, some search mechanisms consider tables as visual objects. Second, tables may be difficult to read, even for the typical user, or at all impossible, for the visually impaired users.

Indeed, the third principle of the W3C recommendation [3] indicates that “*Information and the operation of user interface must be understandable.*”, and item 4 of the Google Web page optimization guidelines [9] recommends that “*(Web page) con-*

tent should be: easy-to-read; organized around the topic; use relevant language; be fresh and unique; be primarily created for users, not search engines". This recommendation reflects the fact that Web crawlers may interpret words randomly or repeatedly distributed in a Web page as an attempt to manipulate page rank, and thereby reject indexing the page.

Finally, we observe that some of the W3C specific recommendations for the visually impaired user in fact coincide with Google's orientations. Comparing the two, it is clear that the difficulties faced by the visually impaired user are akin to those a search engine suffers during the data collection step. As an example, both Google and W3C recommend using the attribute "alt" to describe the content of an image. Naturally, the content of an image is opaque to both a visually impaired user and a search engine, but an alternate text describing the image can be indexed by a search engine and read (by a screen reader) to the visually impaired user. In general, many W-Ray strategies defined to address the limitations of search engines also apply to the design of a database interface for the visually impaired user.

2.2 Guidelines for view design

The designer should first select which data should be published with the help of database views. We offer the following simple guidelines that the designer should follow:

- Attributes whose values have no semantics outside the database should not be directly published.
- Artificially generated primary keys, foreign keys that refer to such primary keys, attributes with domains that encode classifications or similar artifacts, if selected for publication, should have their internal values replaced by their respective external definitions. For example, a classification code should be replaced by the corresponding classification term.
- Attributes that contain private data should not be published.
- Views should not contain too many attributes; only those attributes that are relevant to help locate the objects and their relationships should be selected.

2.3 Translating the materialized data to natural language sentences

The heart of the W-Ray approach lies in the translation of materialized view data to natural language sentences. Fuchs et al. [8] propose a single language for machine and human users, basically by translating English sentences to first-order logic. Others propose to translate RDF triples to natural language sentences [7, 13], simply by concatenating the triples. Tools to translate conventional data to RDF triples have also been developed [2, 6], which typically map database entities to classes, attributes to datatype properties, and relationships to object properties. The proposals introduced in [7, 13] do not consider sequences of RDF triples, though, which we require to compose simple sentences into more complex syntactical constructions. Therefore, we combine the strategies to synthesize sentences described in [13] with the mapping of conventional data to RDF triples introduced in [2].

The translation of materialized view data to natural language sentences involves two tasks: choice of an appropriate *external vocabulary*; and definition of *templates* to guide the synthesis of the sentences.

First observe that the database schema names, including view names, are typically inappropriate to be externalized to the database users. This implies that the designer must first define an *external vocabulary*, that is, a set of terms that will be used to communicate materialized view data to the users. The designer should obey the following generic guideline:

- The external vocabulary should preferably be a subset of a controlled vocabulary covering the application domain in question, or of a generic vocabulary, such as that of an upper-level ontology or Wordnet.

If followed, this guideline permits defining hyperlinks from the terms of the external vocabulary to the terms of the controlled vocabulary. A similar strategy to synthesize sentences is discussed in [11]. An extension to Wordnet is also proposed in [23] to treat concepts corresponding to compound nouns.

After selecting the external vocabulary, the designer must define templates that will guide the synthesis of the sentences. We offer three alternatives: *free* template definition; *default* template definition; and *modifiable default* template definition. The first alternative leaves template definition in the hands of the designer and, thus, may lead to sentences with arbitrary structure. In the default template alternative, the designer first creates an entity-relationship model that is a high-level description of the views, and then uses a tool that generates default templates based on the ER model and synthesizes sentences with a regular syntactical structure. The last alternative is a variation of the second and allows the designer to alter the default templates.

For the free template definition alternative, we offer the following guidelines:

- A template must use the external vocabulary and other common syntactical elements (articles, conjunctions, etc.) [19], as well as punctuation marks.
- A template should generate a sentence that characterizes an entity through its properties and relationships.
- The subject of the sentence should have a variable associated with an identifying attribute of the view.
- The predicate of the sentence should have variables associated with other view attributes that further describe the entity, or that relate the entity to other entities.

The use of free templates is illustrated in what follows, using a relational view of the SIDRA database, which the Brazilian Institute of Geography and Statistics (IBGE) publishes on the Web with the help of HTML forms. The full details can be found in [22].

We start by defining views over the SIDRA database. To save space, Table 1 shows just the “political_division” view: the first column indicates the view name, the second column indicates the attribute names of the view, the third column describes the attributes, and the fourth column associates a variable with each attribute.

We then define a template to publish the “political division” view data:

U is a “*L*” that has a total of *VM* for the year *Y* and aggregate variable *A*.

Table 1 – Schematic definition of a view over the SIDRA database.

View Name	Attribute Name	Attribute Description	Variable
political_division	name	<i>name of the political division</i>	<i>U</i>
	level	<i>level of the political division, such as state, county,...</i>	<i>L</i>
	aggreg_var	<i>name of an aggregation data, such as resident population</i>	<i>A</i>
	aggreg_var_value	<i>value of the aggregation data</i>	<i>V</i>
	unit_measure	<i>unit measure of the aggregation data</i>	<i>M</i>
	year	<i>year the aggregation data was measured</i>	<i>Y</i>
...			

Next, the view is materialized. Each line of the resulting table is transformed into a sentence, using the template. The following sentence illustrates the result:

Roraima is a unit of the federation that has a total of **395.725 people** for the year **2007** and aggregate variable "**resident population**".

Note that: the underlined words are the subject of the sentence; the predicate "is a unit of the federation" qualifies the subject; the words in boldface are view data that play the role of predicatives of the subject, together with the fragments in italics.

We now repeat the example using the default templates alternative. Recall that, in this alternative, the designer starts by creating an ER model of the views. In our running example, the ER model would be:

```
entity(political_division,name).
attribute(political_division,level).
attribute(political_division,aggreg_var).
attribute(political_division,aggreg_var_value).
attribute(political_division,unit_measure).
attribute(political_division,year).
```

Using the variables defined in Table 1, the tool generates default templates such as:

```
'There is a political division with name P'
'The level of P is L'
```

Using default templates, the tool then synthesizes sentences such as (data in boldface):

```
'There is a political division with name Roraima'.
'The level of Roraima is unit of the federation'.
```

Finally, the modifiable default template alternative allows the designer to alter the default templates. Examples of template redefinitions are (where the variables in boldface italics in the new template have to occur in the default template):

```
Default template: 'There is a political division with name P'
New template: 'P'

Default template: 'The level of P is L'
New template: 'is a L'
```

The designer is also allowed to compose the modified templates as in the example:

```
facts((political_division(P),level(P,L)).
```

Using modified templates, the tool synthesizes sentences such as (data in boldface):

```
'Roraima is a unit of the federation'
```

2.4 Guidelines for publishing the sentences as static Web pages

As mentioned before, W-Ray follows the W3C recommendation [3], as well as the Google Web page optimization guidelines [9].

Briefly, the most relevant criteria that W-Ray adopts to publish Web pages are:

- Create hyperlinks between the published data and metadata (W3C Recomm. 3).
- Create hyperlinks between the published data to improve data exploration via navigation (W3C Recomm. 1.3.2 and 2.4 and Google Recomm. 3 and 5).
- Create content with well-structured sentences, as addressed in Section 2.2 (W3C Recomm. 3 and Google Recomm. 4).
- Use text to describe images when the attribute “alt” does not suffice (W3C Recomm. 1.1.1 and Google Recomm. 7).

In the example of Section 2.3, the subject of the sentence – Roraima – would be hyperlinked to a Web Page with further information about the State of Roraima. Briefly, the URLs would be generated upfront by concatenating a base URI with the primary key of the data (see[22] for the details).

3 W-Ray for geographical data in vector format

We first observe that a number of tools [17] offer facilities to convert geographic data in vector format to dynamic Web pages. However, such Web pages are typically not indexed by search engines. We also observe that geographic data in vector format is not opaque, as raster images are, since the data is often associated with conventional data and, in fact, with the (geographic) objects stored in the database. A solution to make vector data visible to the search engines would therefore be to publish the conventional data associated with them, as discussed in Section 2. This strategy would however totally ignore the geographic information that the vector data capture.

In the W-ray strategy, we explore how to translate the relevant geographic information again as natural language sentences. On a first approximation, the strategy is the same as for conventional data: define a set of database views that capture which data should be published; materialize the views; translate the materialized data to natural language sentences; and publish the sentences as static Web pages.

More specifically, suppose that the vector data is organized by layers. Then, when defining a view, the designer essentially has to decide:

- Which layers will be combined in the view. For example, the view might combine the political division, populated places and waterways layers;
- For each layer included in the view, which objects will be retained in the view. For example, one might discard all populated places below a certain population;
- For each layer included in the view, which attributes will be retained in the view;
- When the view combines several layers,
 - Which is the priority between the layers. For examples, the populated places layer may have priority over the political division and the waterways layers;

- Which topological relationships between the objects of different layers should be materialized. For example, for each populated place (of the highest priority layer), one might decide to materialize which navigable waterways (of the lowest priority layer) are within a buffer of 100km centered in the populated place.
- In which topological order the objects will be described. For example, populated places might be listed from north to south and from west to east.

As for conventional data, the designer should select the external names preferably from a controlled vocabulary such as the ISO19115 Topic Categories [12].

For example, consider a view consisting of three layers - the political division, the populated places and the waterways of Brazil - filtered as follows:

- political division: keep only the states, with their name, abbreviated name, area and population, located in the north region
- populated places: retain only the county and state capitals, with their name, political status, area and population, located in the states in the north region
- waterways: keep only the name, navigability and flow

Furthermore, assume that the topological relationship between populated places and political division is ‘*is located in*’ and that between waterways and political division is ‘*cross*’. Assume that populated places have priority and that they are listed from north to south and from west to east.

Examples of sentences would be (using the same conventions as in Section 2.3):

Roraima is a unit of the federation that has a total of **395.725** people for the year **2007** and aggregate variable “**resident population**”. **Roraima** is located in the **North Region**, with an area of **22,377,870** square kilometers.

Boa Vista is a city that has a total of **249.853** people for the year **2007** and aggregate variable “**resident population**”. **Boa Vista** is located in the unit of federation **Roraima** and is the capital city of the **unit of federation Roraima**, with an area of **5,687** square kilometers.

Amazonas is a waterway that crosses the **unit of federation Amazonas** and the **unit of federation Pará**, with flow **permanent** and navigability **navigable**.

The subject of each sentence (underlined words) would also have a hyperlink to a dynamic Web page with the full information about the state or the city, generated by executing a query over the underlying database.

Using default templates, the running example would be restated as follows:

- Declaration of the entity-relationship model:

```
entity(political_division,name) .
entity(populated_places,name) .
entity(waterways,name) .
attribute(political_division,population) .
attribute(political_division,abbreviated_name) .
attribute(political_division,area) .
attribute(populated_places,level) .
attribute(populated_places,local_area) .
attribute(populated_places,local_population) .
attribute(waterways,flow) .
```

```

attribute(waterways, navigability).
relationship(located_in, [populated_places, political_division]).
relationship(crosses, [waterways, political_division]).

```

- Examples of synthesized sentences, using default templates (with data in boldface):

```

'There is a populated places with name City of Boavista'.
'There is a political division with name State of Amazonas'.
'There is a political division with name State of Pará'.
'There is a waterways with name Amazon River'.
'The flow of Amazon River is permanent'.
'The navigability of Amazon River is navigable'.
'City of Boavista is related to State of Roraima by located in'.
'Amazon River is related to State of Amazonas by crosses'.
'Amazon River is related to State of Pará by crosses'.

```

Turning to the modified default templates alternative, examples are:

- Template redefinition:

Default template: 'There is a political division with name **P**'

New template: 'The **P**'

Default template: '**R** is related to **P** by crosses'

New template: 'is crossed by **R**'

Default template: 'The flow of **R** is **F**'

New template: 'which is **F**'

Default template: 'The navegability of **R** is **V**'

New template: 'and **V**'

- Template composition:

```

facts((political_division(P), crosses(R, P),
      flow(R, S), navigability(R, V)).

```

- Sentences generated using the new templates (with data in boldface):

```

'The State of Amazonas is crossed by Amazon River which is permanent
and navigable'
'The State of Pará is crossed by Amazon River which is permanent
and navigable'

```

4 W-Ray for raster data

Following the idea introduced in Leme et al. [14], the W-Ray strategy describes raster data by publishing sentences that capture the metadata describing how the raster data was acquired, and the geographic objects contained within its bounding box.

The geographic objects might be obtained, for example, from a gazetteer, such as the ADL gazetteer [10], which includes a useful Feature Type Thesaurus (FTT) for classifying geographic features. As for vector data, the designer should define views, this time based on the classification of the geographic objects.

As a concrete example, consider the image fragment of the City of Rio de Janeiro, taken out of the Web site “Brazil seen from Space”, and assume that:

- the metadata of the image indeed indicates the coordinates of its bounding box

- the geographic objects and their classifications are taken from the ADL Gazetteer
- the designer decides to associate images with geographic objects classified as ‘hydrographic feature’, a topic category of FTT, whose centroid is contained in the bounding box of the image

The raster image would then be processed as follows:

1. The georeferencing parameters are extracted from the image. In this case, the image fragment is consistent with a scale of 1:25.000 and has bounding box defined by $((43^{\circ}15'W, 22^{\circ}52'30''S), (43^{\circ}07'30''W, 23^{\circ}S))$.
2. By querying the ADL Gazetteer using the georeferencing parameters extracted in Step 1 and the ADL FTT term selected, ‘hydrographic feature’, one locates 9 objects, which the first few are:
 - a. *Feature*(“*Rodrigo de Freitas, Lagoa - Brazil*”, *lakes, contains*)
 - b. *Feature*(“*Comprido, Rio – Brazil*”, *streams, contains*)
 - c. *Feature*(“*Maracana, Rio – Brazil, streams, contains*)

The query results would be translated to the following sentence, describing the image (using the same conventions as in Section 2.3):

The image of Rio de Janeiro, Brazil, contains the lake “**Rodrigo de Freitas**” and the streams “**Comprido**” and “**Maracanã**”.

where the underlined words form the subject of the sentence, the words in boldface italics were extracted from the ADL FTT, and those in boldface denote geographic objects in the ADL Gazetteer whose centroids are contained in the bounding box of the image.

5 Conclusions

This paper outlined an approach to overcome the problem of accessing conventional and geographic data from the Deep Web. The approach relies on describing the data through natural language sentences, published as Web pages. The Web pages thus generated are easily indexed by traditional search engines, but they also facilitated the task of engines that support semantic search based on natural language features. The details of the approach can be found in [22].

Further work is planned to assess which of the three alternatives for generating templates, if any, leads to better recall. The experiments will use massive amounts of data from geographic databases organized by IBGE, as well as a large multimedia database.

Lastly, we remark that the approach can be easily modified to generate RDF triples, instead of natural language sentences, and to cope with multimedia data. In a broader perspective, it can also be used to describe conventional, geographic and multimedia data to the visually impaired users. The challenges here lie in structuring the sentences in such a way to avoid cognitive overload.

Acknowledgements. This work was partly supported by IBGE, CNPq under grants 301497/2006-0, 473110/2008-3, 557128/2009-9, FAPERJ E-26/170028/2008, and CAPES/PROCAD NF 21/2009.

References

- [1] BERGMAN, M. K. 2001. The Deep Web: Surfacing Hidden Value. *J. Electr. Pub.* 7(1).
- [2] BIZER, C. and CYGANIAK, R., 2006. D2R Server – Publishing Relational Databases on the Web as SPARQL Endpoints. In *Proc. 15th Int'l. WWW Conf.*, Edinburgh, Scotland.
- [3] CALDWELL, B.; COOPER, M.; REID, L.G. and VANDERHEIDEN, G. 2008. Web Content Accessibility Guidelines (WCAG) 2.0. In *W3C Recommendation*.
- [4] CALLAN J. 2000. Distributed information retrieval. In *Advances in Information Retrieval*, Eds. Springer, US, 127-150.
- [5] COSTA L. 2005. Esfinge - Resposta a perguntas usando a Rede. In *Proc. Conf. Ibero-Americana IADIS WWW/Internet*, Lisboa, Portugal.
- [6] ERLING, O. and MIKHAILOV, I. 2007. RDF support in the virtuoso DBMS. In *Proc. 1st Conference on Social Semantic Web*, Leipzig, Germany, Vol. 113 of LNI, pp. 59–68.
- [7] FLIEDL G.; KOP C. and VÖHRINGER J. 2010. Guideline based evaluation and verbalization of OWL class and property labels. *Data & Knowledge Eng.* 69(4), pp. 331-342.
- [8] FUCHS N. E.; KALJURAND K. and KUHN T. 2008. Attempto Controlled English for Knowledge Representation. In *Reasoning Web 2008*, LNCS 5224, pp. 104-124.
- [9] GOOGLE. 2008. In *Google's Search Engine Optimization Starter Guide*, Version 1.1.
- [10] Alexandria Digital Library, 2004. Guide to the ADL Gazetteer Content Standard, v. 3.2
- [11] HOLLINK, L.; SCHREIBER, G.; WIELEMAKER, J. and WIELINGA, B. 2003. Semantic Annotation of Image Collections. In *Proc. Knowledge Markup and Semantic Annotation Workshop*, Sanibel, Florida, USA.
- [12] ISO 19115:2003, Geographic Information – Metadata.
- [13] KALYANPUR A.; HALASCHEK-WIENER C.; KOLOVSKI V. and HENDLER J. 2005. Effective NL Paraphrasing of Ontologies on the Semantic Web. In *Workshop on End-User Semantic Web Interaction, 4th Int. Semantic Web conference*, Galway, Ireland.
- [14] LEME L. A. P. P.; BRAUNER D. F.; CASANOVA M. A. and BREITMAN K. 2007. A Software Architecture for Automated Geographic Metadata Annotation Generation. In *Proc. XXII Simpósio Brasileiro De Banco De Dados, SBBD*, João Pessoa, Brazil.
- [15] MADHAVAN J.; AFANASIEV L.; ANTOVA L. and HALEVY A. 2009. Harnessing the Deep Web: Present and Future. In *Proc. 4th Biennial Conf. on Innovative Data Systems Research (CIDR)*, Asilomar, California, USA.
- [16] MADHAVAN, J.; KO, D.; KOT, L.; GANAPATHY, V.; RASMUSSEN, A. and HALEVY, A. 2008. Google's Deep-Web Crawl. In *Proc. VLDB* 1(2), pp. 1241–1252.
- [17] MapServer. <http://mapserver.org/about.html#about>
- [18] MENG W.; YU C.T. and LIU K.L. 2002. Building efficient and effective metasearch engines. *ACM Computing. Survey*, v. 34, n.1, pp. 48-89.
- [19] PRANINSKAS, J. 1975. *Rapid review of English grammar*. Prentice-Hall, NJ, USA.
- [20] RAGHAVAN S. and GARCIA-MOLINA H. 2001. Crawling the HiddenWeb. In *Proc. VLDB*, pp. 129-138.
- [21] RAJARAMAN A. 2009. Kosmix: HighPerformance Topic Exploration using the Deep Web. In *Proc. VLDB, Lyon, France*.
- [22] PICCININI, H.; LEMOS, M.; CASANOVA, M.A.; FURTADO, A.L. 2010. W-Ray: A Strategy to Publish Deep Web Geographic Data. Tech Rep. 10/10. Dept. Informatics, PUC-Rio.
- [23] SORRENTINO S.; BERGAMASCHI S.; GAWINECKI M. and PO L. 2009. Schema Normalization for Improving Schema Matching. In *Proceedings of the 28th International Conference on Conceptual Modeling- ER*, Gramado, Brazil, LNCS 5829, pp. 280-293.
- [24] ZHENG, Z. 2002. AnswerBus question answering system. In *Proc. 2nd International Conference on Human Language*, San Diego, California, pp. 399–404.