

Semantic Extraction of Geographic Data from Web Tables for Big Data Integration

Isabel F. Cruz
ADVIS Lab
Department of Computer
Science
University of Illinois at Chicago
ifc@cs.uic.edu

Venkat R. Ganesh
ADVIS Lab
Department of Computer
Science
University of Illinois at Chicago
vganesh4@uic.edu

Seyed Iman Mirrezaei
ADVIS Lab
Department of Computer
Science
University of Illinois at Chicago
smirre2@uic.edu

ABSTRACT

There are millions of web tables with geographic data that are pertinent for big data integration in a variety of domain applications, such as urban sustainability, transportation networks, policy studies, and public health. These tables, however, are heterogeneous in structure, concepts, and metadata. One of the challenges in semantically extracting geographic data is the need to resolve these heterogeneities so as to uncover a conceptual hierarchy, metadata associated with instances, and geographic information—corresponding respectively to ontologies, elements that we call *features*, and cell values that can be used to identify geographic coordinates. In this paper, we present an architecture with methods to: (1) extract feature-rich web tables; (2) identify features; (3) construct a schema and instances using RDF; (4) perform geocoding. Preliminary experiments led to high accuracy in table identification and feature naming even when compared to manual evaluation.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Spatial databases and GIS*; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Query formulation*

General Terms

Algorithms, Experimentation

Keywords

Geographic data, Web tables, Information extraction, Semantic data integration, Geocoding, Spatial databases, GIS

1. INTRODUCTION

We have a vision in which domain experts in urban sustainability, transportation networks, policy studies, and public health extract and integrate geographic data from several web pages to build links among data so as to be able

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
GIR'13, November 05 2013, Orlando, FL, USA

Copyright is held by the owner/author(s).

Publication rights licensed to ACM.

ACM 978-1-4503-2241-6/13/11\$15.00.

<http://dx.doi.org/10.1145/2533888.2533939>

to derive complex interrelationships. However, there is a large gap between this vision and reality: the web contains millions of tables [7]. Several of these web tables are published as reports or historic data by organizations such as the U.S. Bureau of Transportation Statistics,¹ CityOfChicago,² and Pearson Education.³ We refer to these tables that contain high-quality relational data as *feature-rich*. These data, which provide information about real-time events, are usually described using three important dimensions—spatial, temporal and thematic [20]. Although these tables expose a wide range of information, they are often confined to a particular theme depending on the purpose of a website. For example, a table on a website dedicated to “Precipitation in Chicago” may show the monthly average rainfall/snowfall specific to a Chicago subregion. Further, many of these tables are manually created. Their purpose is to be interpreted by humans not by machines and, quite often, their underlying databases (when they exist) cannot be web queried.

However, automatic extraction is needed. For example, consider another website containing data on “Disease statistics in Illinois” showing the influenza daily statistics in various cities. Here, the disease statistics may be reported in percentage with a column header like *Statistics (%)* or the date in *M/D/Y* format. We refer to these implicit metadata elements as *features*. A domain expert could examine the tables to understand how the different values and features are linked and then proceed to perform the correlation between precipitation and influenza values in Chicago. While this approach may sound feasible for two tables, the complexity of relating a large number of tables manually makes the task nearly impossible. Thus, we would like to develop an automatic system that can uncover the semantic links that relate data in those tables.

We refer to the process of linking data by automatically identifying and mapping the underlying semantics as *data integration*. In our example, the only data point present in both tables is the geographic component—Chicago—and even this common point may or may not be directly perceived, depending on how the connection to Chicago is established. For example, some data may have been collected at the level of wards, while other data may come from a sensor network placed with no connection to the city’s administrative units.

Apart from spatial heterogeneity, tables may have different temporal resolution (monthly, daily) and different con-






¹<http://www.rita.dot.gov/bts>

²<https://data.cityofchicago.org/>

³<http://www.infoplease.com>

[Home](#)
[World](#)
[U.S.](#)
[Homework Help](#)
[People](#)
[History & Gov't](#)
[Science & Health](#)
[Calendar & Holidays](#)
[Business](#)
[The Fifty States](#)


[Weather](#)

Climate of 100 Selected U.S. Cities

The following table gives the average monthly temperature, average annual precipitation, and average annual snowfall for 100 cities in the United States.

City	Average monthly temperature (°F) ¹				Precipitation		Snowfall ²	
	Jan.	April	July	Oct.	Average annual		Number of years observed ⁴	
					(in.) ¹	(days) ³		Average annual (in.) ³
Albany, N.Y.	22.2	46.6	71.1	49.3	38.60	136	64.4	57
Albuquerque, N.M.	35.7	55.6	78.5	57.3	9.47	60	11.0	64
Anchorage, Alaska	15.8	36.3	58.4	34.1	16.08	115	70.8	39 / 60
Asheville, N.C.	35.8	54.1	73.0	55.2	47.07	126	15.3	39



like the look, love the savings you gotta go to Ross

find your store >

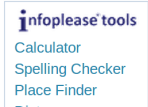




Figure 1: Screenshot of a web page that includes a table.

cepts (precipitation, influenza) or the same concept and different names (influenza, flu). To facilitate information extraction that takes time, space, and other attributes into account, with the goal of integrating data to allow for geospatial queries and for data visualization on a map, a well-founded and robust methodology is needed.

We note that some research problems that are apparently related to this paper are in fact quite distinct. In one of them, behind the tabular data representation, there are databases. The tabular data result from queries made through forms to those databases. The goal is to use those forms as gateways to the databases so as to crawl the so-called “deep web” [19]. Other work considers templates for shopping web sites, which have a “consistent” (even if heterogeneous) display of the hierarchies of products (and of their attributes) on those web sites [25]. In our work, domain databases and forms are often absent and common templates are hard to find. Other work directed to shopping sites identifies semi-structured tables using a classification algorithm [8] instead of templates that extract attribute/value pairs [25]. Because of their use of a classification algorithm, there are similarities with our work, yet there are also important differences because the shopping and geospatial domains differ substantially.

In this paper, we present several elements of an architecture for semantically extracting geographic data from web tables. Such elements provide the following functionalities:

- Automatic identification of feature-rich web tables.
- Automatic feature tagging within the tables, including the tagging of measurement units and of temporal data.
- Ontology construction from table captions and complex tables with nested headers.
- Geocoding [28] with disambiguation methods using the constructed ontology.

The main goal of our information retrieval task is to produce appropriate data in a semantic web format to facilitate both geospatial and semantic queries on them. This work is a part of our larger semantic framework *GIVA*, which com-

bines geographic data extraction, geospatial and temporal data integration, visualization, and analytics [10].

The rest of this paper is organized as follows: Section 2 explains the features and characteristics of typical web tables that contain geographic information. In Section 3, we overview the architecture of our system. Section 4 describes the preliminary implementation of the main architectural components of our system and presents experimental results. In Section 5, we make several observations and outline future work. Section 6 covers related work on semantic data-extraction from tables and place name disambiguation in geocoding. Finally we draw some conclusions in Section 7.

2. WEB TABLES

A screenshot of a web page with a feature-rich table is shown in Figure 1. We use this example throughout this section to describe various characteristics of a web table.

2.1 Uses

Web tables are not only used for representing relational (feature-rich) data but for various other purposes [7]. For instance, in Figure 1, the `<table>` tag is used to place the horizontal menu bar at the top, advertisements at the right, and the actual feature-rich table in the center contributing to a total of three tables.

2.2 Heterogeneity

Tables exhibit heterogeneity in several ways [9]. In our domain, heterogeneity is mainly due to the presence of nested headers that inherently possess multiple features. We consider the following three heterogeneity types.

Structural. Most web tables have a table header (defined by `<th>`) and table columns (defined by `<td>`). Both of these are created inside a table row (defined by `<tr>`). If table headers are defined in a single row:

```

<tr>
  <th>City</th>
  <th>State</th>
</tr>

```

then this representation is similar to that of a table in a relational database. Recovering the semantics from such tables is an easy task especially when the headers are well defined. Several tools such as D2R [6], Triplify [2], or Karma [21] perform this operation on tables of a relational database. However, most of the web tables have a blend of simple headers and nested headers. For example, in Figure 1, the first column *City* has a simple structure while the second and third columns define a hierarchy by means of nested cells. We refer to such differences as structural heterogeneity in web tables.

Conceptual. Although a web table can be confined to a specific theme and purpose, there are also situations where data about multiple concepts are described in the same table. Figure 1 shows this heterogeneity by displaying three different concepts, namely *Average monthly temperature*, *Precipitation*, and *Snowfall*, in a single table.

Metadata. Metadata is defined as data about data. In web tables, it could be any information that summarizes the entire table (table caption) or could be the information present in the table headers that describes data in the respective columns. Usually, every table has a caption associated with it, generally placed above it. However, identifying the caption automatically is challenging due to the variations in web page layouts. For example, in Figure 1, the table caption *Climate of 100 Selected U.S. Cities* is not present immediately above the table but a few lines above it. The table header is also an important source of metadata. That is, metadata is often implicit whether in the caption or present elsewhere in the web page. We refer to the implicit metadata elements as *features*, which are further discussed next.

2.3 Features

Table headers contain a variety of features. We will discuss here some of the important features that are useful for geographic information retrieval. Table 1 shows a few examples of features available in table cells.

Spatial	Albany, N.Y.; Anchorage, Alaska
Temporal	Jan.; Oct.
Units	°F; (in.); (days)
Misc	Snowfall ²

Table 1: Some features present in the web table shown in Figure 1.

Spatial. Many web tables that contain geographic information have data in one or more columns with location information, such as cities, parks, or counties. This kind of information does not make sense without identifying the associated metadata. For instance, the word “Cook” is recognized as a county (in Illinois) only if the respective column header is properly identified. There are also situations where city and state names are combined in a single cell as in “Albany, N.Y.” introducing challenges in data extraction.

Temporal. Temporal data represented in HTML tables can be specific (02/11/2013 01:24:03), approximate (Oct, 2012), or a range (May-June 2013). Approximate temporal data can also be represented as a temporal range. For example, October 2012 represents the range October 1-31, 2012. This kind of interpretation is required when performing temporal queries.

Units. Any historic data or reports that involve numbers

will have measurement units, which may be encoded in the table headers as shown in Figure 1 or within the data cells such as \$12,000 or 1.2in.

Ambiguities. Ambiguities are common in geographic data extraction. For example, in Figure 2, “Lincoln Park” is an ambiguous location that may refer to a neighborhood in Chicago or Albany. Superscripted symbols (or numbers) may also introduce ambiguity. For instance, the number “2” in the unit m² (square meter) is different from the same superscript in Snowfall².

3. ARCHITECTURE

In this section, we propose an architecture for our system for the semantic extraction of geographic data from web tables and describe briefly its components. The architecture is shown in Figure 2. In this section we will be making use of NLP techniques. In what follows, terminology that represents tags and patterns (e.g., DT, NN[*], JJ[*], VBP, PP NN) is taken from a well-known Part-Of-Speech (POS) tagging technique [26].

3.1 Table Extraction

To identify feature-rich tables, we use a decision tree classifier with features listed in Table 2. This model is then trained using 500 heterogeneous web tables. Each feature-rich table is given a unique *Table ID* and its position in the original web page is marked using a label. This position information will be later used to identify table captions as described in Section 3.3.

Feature	Value
Number of Columns	S(1-3), M(4-7), L(>7)
Number of Rows	S(1-5), M(6-10), L(>10)
Number of , <object> tags in tables cells	S(1-3), M(4-10), L(>10)
Background color difference between the rows	True, False
Font weight difference between the rows	True, False
Presence of <th> in first two rows	True, False

S, M, L = Small, Medium, Large

Table 2: Decision tree features for table extraction.

3.2 Preprocessing

A feature-rich table is preprocessed in order to convert it into a *table matrix*. The rows of this matrix that correspond to the table headers are referred to as *header cells* and the rest as *data cells*. To create this matrix, we first identify the table headers using the <th> tags. The initial matrix is generated by expanding *rowspan* and *colspan* tags. Then, a POS tagger runs on the *data cells*. The cells that contain multiple concepts (noun groups [NN*] separated by commas) are split to form a final matrix as shown in Figure 3. Because of this expansion, the dimension of this matrix may not be the same as that of the original table. We use the *table matrix* throughout our architecture for data extraction.

3.3 Feature Extraction

This component identifies several features as described in Section 2.3 and labels them appropriately using NLP techniques.

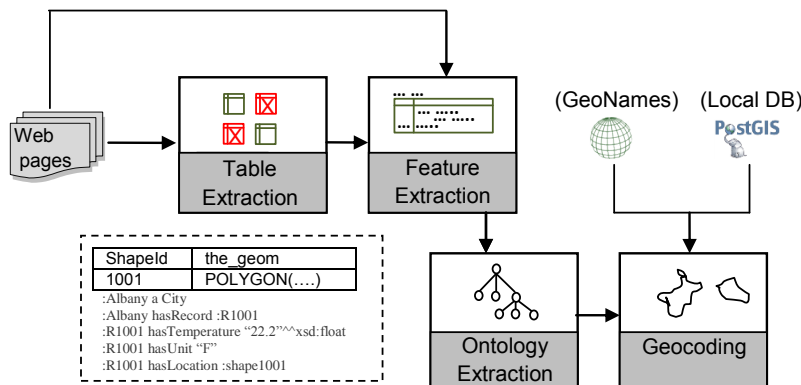


Figure 2: System architecture.

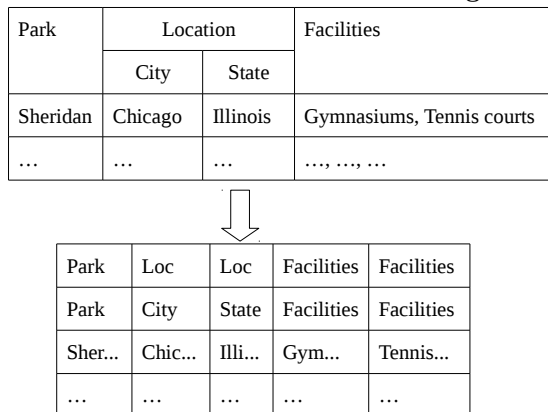


Figure 3: Creation of the *table matrix*.

3.3.1 Caption

In order to capture the table caption from a web page, we use a greedy approach on the most frequent HTML style tags that are used to define a caption. The text content above the web table is scanned bottom-up. The first identified text block with the following HTML tag priority is taken as the caption: (1) `<h1>` to `<h6>`; (2) HTML style tags (e.g., ``, `<i>`); (3) `` with font related CSS-styles. If no caption is identified at this stage, then the web page title (defined using the `<title>` tag) is taken as the caption. This caption is then annotated to identify the POS tags. Predefined phrases such as *Table 1:* and *This table explains*, English articles (DT) and symbols are removed. The remaining text becomes the caption, which is a noun group (NN[*]) with or without adjectives (JJ[*]).

3.3.2 Location

To identify a column containing location information, we first perform entity recognition on the header cells using a custom dictionary. This dictionary is created using the feature codes from GeoNames.⁴ This mechanism assists in the identification of headers such as those that describe *States* or *Parks*. However, such clear headers may be often unavailable. To resolve this, we perform POS tagging on random rows of *data cells* to identify only the noun groups. These words are used to extract a maximum of ten sample sentences from 13 million English pages of Wikipedia, which we have indexed locally. The sentences then become an in-

⁴<http://www.geonames.org/export/codes.html>

put to a Named Entity Recognizer (NER) that annotates a data cell as a location, person, date, etc. We use a label *LOC* to indicate the *data cells* that contain location information. While it is true that this method performs accurately only on a well phrased sentence, it still helps us to separate location data from the rest. For example, a flight status web table may contain a *Status* column with words such as “Delayed” or “On time” in the data cells. This column can easily be eliminated as it would be annotated as a verb tense (VBP) and a preposition followed by a noun (PP NN), respectively.

3.3.3 Measurement units

Units are identified by creating templates that use a dictionary of measurement units [35]. These templates are applied to the cells and the units are tagged and separated from the numeric values appropriately. A label *UNIT* is assigned to the appropriate cell along with a link to the dictionary item. For example, the cell value “\$12,000” will be annotated as “UNIT{ \$ }, VALUE{ 12000 }”.

3.3.4 Temporal data

We use custom templates to identify and translate the temporal data into a string. An assumption that we make is that the date is in M/D/Y format and the first three components of the time are H:M:S. The identified temporal information is converted into a temporal range of GMT strings with begin and end date. For instance, Oct, 2012, will be assigned a label TIME: 10/01/2012 00:00:00 to 10/31/2012 23:59:59. Default values are used; for example, the beginning of a month is its first day at time 00:00 unless otherwise specified.

3.4 Ontology Extraction

Ontologies play a key role in the semantic realization of data and for facilitating heterogeneous data integration [4]. We use RDF/S to express our ontologies. We take advantage of most of RDF Schema properties to construct an ontology. We extract the hierarchy and data from the table caption and *table matrix* as described below.

3.4.1 Schema

The RDF schema is constructed using the table caption and the header cells of the *table matrix*. The default classes and properties that will be present in the schema are listed in Table 3.

The *Geometry* class will take the data type *rdfs:Literal*. This class will be populated with locations as described

Classes: Geometry, Date, BeginDate, EndDate, Unit Properties: hasGeometry, hasBeginDate, hasEndDate, hasUnit :BeginDate rdfs:subClassOf :Date :EndDate rdfs:subClassOf :Date

Table 3: Default classes and properties.

in Section 3.5. The *Date* class will take the data type *xsd:dateTime* to represent a temporal range. Apart from this, all classes will have *rdfs:comment* that will be populated as described in later sections. We do this to facilitate more accurate data integration using our ontology mapping system AgreementMaker [11, 15].

Caption. We construct a hierarchy from the table caption using a priority set of POS rules as listed in Table 4. The symbol “→” indicates *immediately follows*. The noun group (NN[*]) is used to construct the class name. An *rdfs:label* is assigned for a human readable version of the class name. The table URL along with a *Table ID* will become the URI. To populate *rdfs:comment*, we use our local Wikipedia repository to search and extract the first two sentences from the *Introduction* section of Wikipedia pages as a comment, provided there are no ambiguities. A *has* relationship property is assigned between two resources based on the hierarchy.

Node	POS Rule	Hierarchy
1	IN → NN[*]	Root
2	JJ → NN[*]	Child of Node 1
3	TO → NN[*]	Child of Node 2
4	Default	Root

Table 4: POS rules to identify a hierarchy from a table caption.

Header cells. The hierarchy of the associated table is formed by looking at how the header cells are arranged. The features labeled in these cells are used along with the hierarchy to construct a schema using *is-a* relationships [14]. Based on the position of *IN* and *TO*, a class name generation takes place. For instance, the class name for *Number of years observed* will be *Observed_Years*. The empty header cells are replaced with names based on the corresponding data cell labels. For instance, if the data cells are labeled as *LOC* then the header cell falls under the class *Location*. A default class name *Data* is used otherwise. Cells labelled as *UNIT* will populate *rdfs:comment* using measurement dictionary units.

3.4.2 Instances

We use the constructed RDF Schema and create RDF triples using the data cells of the *table matrix*. Because we deal with geographic data and historic reports, every row of a data cell has multiple concepts within it. As an example, Figure 1 contains three different concepts that are temporally distinct. To generate triples, we define a unique *Record ID* and generate a triple for every concept that is shown in the sample triple data of Figure 2.

3.5 Geocoding

The process of identifying the associated geographic coordinates from textual data is referred to as geocoding [28]. This process should inherently possess capabilities to disambiguate locations so as to determine the exact geographic

coordinates. We describe here our methods to identify geographic coordinates and how we disambiguate those coordinates by using the ontologies constructed by our system.

3.5.1 Geographic coordinates

We use two different systems to extract geographic coordinates: GeoNames⁵ and PostgreSQL (with PostGIS⁶ extension). For any location, GeoNames returns *Point* data with an appropriate feature class (e.g., airport, park, country, or mountain), which we refer to as *Geo-Context*. However, because of our focus on big data integration (spatial and temporal), a simple point data for a location will not be sufficient. Any location that is spread across the geography of the globe is a region that may be defined by a polygon or a polyline. For instance, a ZIP code spreads across a region and not just a point. Although such information may be unavailable for all locations on the planet, geographic information such as boundaries of cities, counties, states, ZIP codes and water bodies in the U.S. are made available to the public by the U.S. Census Bureau,⁷ the U.S. Geological Survey (USGS),⁸ and others. We spatially index these data in PostgreSQL. We maintain only four kinds of relevant information in the database: *ShapeId*, *Geometry*, *Name*, and *LocationType*. All data cells labeled as *LOC* in *table matrix* are used for geocoding using GeoNames. If a place is unambiguous, a geospatial query *Contains* is performed in the local database to identify the shape information. The results of this query are also added to *Geo-Context*. Once the entire *table matrix* is processed, the disambiguation process takes place as described in the next section.

3.5.2 Disambiguation

One of the major challenges in geocoding is the disambiguation of a toponym (place name). Context information is necessary for any disambiguation method to work well. Further, Tobler’s first law of geography states that “*Everything is related to everything else, but near things are more related than distant things*” [37]. Thus, distance also becomes an important metric in disambiguation.

We use the context as well as the distance to disambiguate the toponyms as illustrated in Figure 4. The *Table-Context* is a list of RDF labels (*rdfs:label*) retrieved from the constructed schema and a list of all unambiguous geocoded places. We use a similarity matcher from the AgreementMaker ontology matching system, which has the advantage of being extensible [11]. For this concept-based string similarity algorithm a similarity threshold can be set. We run it on top of *Geo-Context* and *Table-Context*. The matcher is modified to use our custom dictionary (see Section 3.3.2) and WordNet [29]. The output of this matcher is a confidence value that ranges between 0 and 1. For matching using the distance metric, we use both *Context* and measure the distance on a map using the geographic coordinates. The distances are always measured between the unambiguous place names (in Figure 4, “Chicago”) and the ambiguous place names from *Geo-Context*. The distances are rescaled to a numeric range between 0 and 1 (1 being the farthest point identified on the map). We use these two results to disambiguate the place names.

⁵<http://www.geonames.org/>

⁶<http://postgis.net>

⁷<http://www.census.gov/geo/www/tiger/>

⁸<http://www.usgs.gov/>

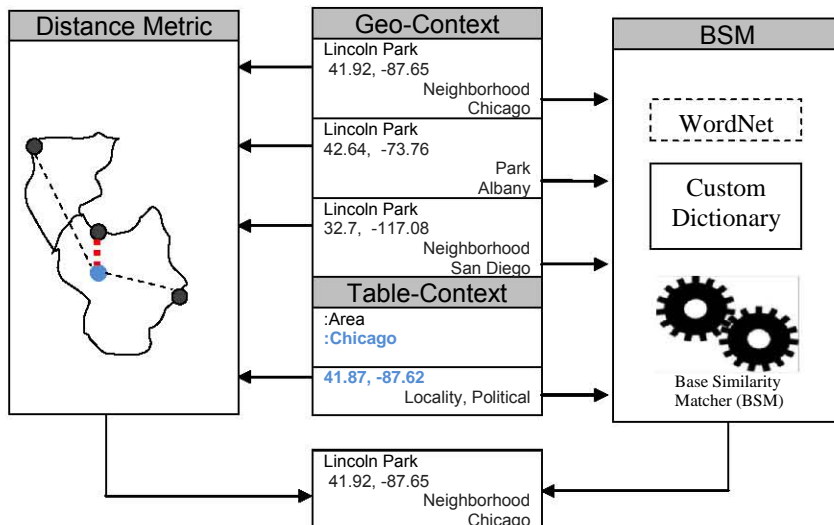


Figure 4: Disambiguation for “Lincoln Park” using Base Similarity Matcher and a distance metric. In a distance metric, the blue point indicates the unambiguous location and the thick red line indicates the shortest distance.

After disambiguation, a triple is created with an appropriate *shapeId* as shown in Figure 2 using the model described in Table 3. If shape information from PostgreSQL cannot be retrieved, the unambiguous point coordinates are stored in the database using a triple.

4. EXPERIMENTS

We use 20 web pages from three different web sites—the U.S. Bureau of Transportation Statistics,⁹ National Oceanographic Data Center,¹⁰ and Pearson Education¹¹—for initial system testing. These web pages contribute to a total of 56 tables of which only 30 are feature-rich. The pages are checked for missing HTML tags and are corrected using HTML Tidy.¹² The tables are extracted using a typical DOM parser and every table is saved as an HTML file. We use WEKA [18], an implementation of the C4.5 [33] decision tree algorithm for table extraction. For all of the NLP based tasks, we use Stanford’s NLP tools¹³ and Jena [27] for ontology extraction.

We measure the accuracy of each component, as shown in Table 5. It is calculated as the *number of correctly identified elements/total number of correct elements*. The table extraction failed to identify one feature-rich table leading to 96.6% accuracy. Feature extraction gave a mean accuracy of 92%. Ontologies were manually evaluated by giving more importance to the constructed hierarchy than to their representation in RDF. Therefore, we increased errors in the following order: (1) invalid hierarchy; (2) improper class names. We do not measure the quality of the class names in relation to the linked open data. This leads to an accuracy of 91%. Geocoding had an accuracy of 85.5%. We also report different results on disambiguation in Table 6. The maximum accuracy was attained when we placed larger weights

for Distance Metric (DM) while combining it with the Base Similarity Matcher (BSM) of AgreementMaker. The baseline accuracy of 40% is achieved using only unambiguous place names retrieved through geocoding. Although this evaluation is made on 30 place names only, we find our disambiguation methods that use ontologies to be working well and therefore worthy of being tested more extensively.

Component		Accuracy (%)
Table Extraction		96.6
Features	Caption	92
	Location	95
	Units of Measurement	89
	Temporal	93
Ontology Extraction		91
Geocoding		85.5

Table 5: Accuracy of the architectural components.

Disambiguation	Accuracy (%)
BSM + DM	85.5
BSM	81.5
DM	78
Baseline	40

Table 6: Geocoding accuracy.

5. DISCUSSION

For the *Table extraction* component, we found that not all tables were constructed using the `<table>` tag. A few tables used `<div>` and some were not formatted properly, thus reducing accuracy. One major assumption we make is that the web tables contain geographic information in at least one column. In the future, we would like to include features to identify tables that contain geographic information through annotations.

⁹<http://www.rita.dot.gov/bts>

¹⁰<http://www.nodc.noaa.gov/>

¹¹<http://www.infoplease.com>

¹²<http://www.w3.org/People/Raggett/tidy/>

¹³<http://nlp.stanford.edu/software>

Feature extraction was particularly challenging when we encountered more complex tables that had lengthy table captions or column headers. For instance, we found the following table header (from www.bts.org) with an unfamiliar unit of measurement: “Distance Shipped (Based on Great Circle Distance)”. Further, there is no description or conversion method associated with the measurement units. We plan to use an ontology for measurement units such as OM [34]. This ontology could help in unit conversions, which can further assist in visualization. We also note that the footnotes marked by a superscripted number/symbol provide useful information. In Figure 1, the superscript “1” contains the text “Years 1971-2000”, which may assist in extracting temporal features.

We found that the performance of the ontology extraction component depends largely on the extent to which the English phrases are well-formed. For instance, our POS rules failed to identify the hierarchy from the table caption because of incorrect tagging. In some cases, the word “like” was tagged as IN (a subordinating conjunction) and VBP (a verb tense) interchangeably. Further, our class name creation does not try to retrieve suggestions from Linked Open Data (LOD) such as DBpedia.¹⁴ However, this drawback can be removed in the future by using AgreementMaker for LOD [12, 13].

Geocoding and *toponym disambiguation* constitute a wide research area. In this paper, we introduced preliminary methods to demonstrate how we combine semantic and geographic data in our architecture and how we use ontologies for disambiguation. We uncovered the need to combine manually multiple columns (e.g., Apartment number, Street and City) for geocoding. Also, disambiguation methods were not accurate in the absence of context and unambiguous place names, which are used for the disambiguation process.

Another future improvement to the system is the conversion of the entire architecture to a single machine learning model. In fact, this was the main reason behind using the *table matrix* as it provides a well defined structure for automatic processing. We also use the scale of 0 to 1 for the matching metric in Section 3.5.2 for this reason. We plan to look into GeoSPARQL¹⁵ and investigate the usage of tools like Parliament [3] into our architecture.

6. RELATED WORK

Quercini and Reynaud [32] address the challenging problem of discovering entities by annotating web tables. They focus on the problem of data extraction from Google Fusion Tables [17] to create a repository of information on points of interest in cities. They identify geographic entity types such as *restaurants*, *cities*, *museums* using a domain ontology and use the annotated data to develop a search system that can answer queries. Their information extraction focus is therefore different from ours; another difference is that the tables they consider are well-structured and contain homogeneous features in a single column. Other related work is by Lieberman et al. [24], which identifies location information from spreadsheets. However, this approach relies on the column headers to identify: (1) whether they have location information; (2) the type of location such as county, city or state. When these headers are missing—which is often the

case with web tables—data extraction will be incomplete.

There has been considerable work performed using semantic data extraction [2, 5, 16, 21, 30, 36, 38] with the aim of publishing linked open data. For instance, Knoblock et al. [21] present a methodology to build data models by looking at the semantics of well-structured data (KEGG pathway¹⁶ data sources) with user interaction. The key contribution in this work is that a better semantic description is achieved by allowing users to refine it manually using a GUI. Similarly, Venetis et al. [38] present a system to extract table semantics by creating a database of class names and relationships and by annotating structured tables. Although both approaches present accurate methods to construct an ontology from tables (by proposing algorithms for class name labeling), they deal with tabular data that are well-structured with easily identifiable and explicit features (e.g., tables from Wikipedia or relational databases). Further, features within tables and geographic data were ignored. There are also tools such as Triplify [2] and D2R [5] that publish RDF triples from relational databases.

Andogah et al. [1] use a list of ranked unambiguous place names from text documents for disambiguation. The documents are first sent to a tagger to recognize place names, organization names, and person names. For each document, the location type (such as continent, country, or city) is identified using unambiguous place names. This identified location type is used to filter other place names that are ambiguous in the document. There are other approaches [22, 23, 31] where the authors use either information about location types or geographic coordinates to apply statistical methods such as Pointwise Mutual Information (PMI) or Log-Likelihood Ratio (LLR) to disambiguate. However, all of them are focused on documents or web pages. One could argue that extracting context information from text documents is somewhat easier than identifying context from a web table.

In summary, while some approaches focus on semantic data extraction and others on geographic data extraction, we have not found other approaches that need to consider both like we do, given our aim to perform geospatial data integration and visualization.

7. CONCLUSIONS

In this paper, we proposed an architecture to extract semantic and geographic data for big data integration. In particular, we described the level of complexity and meta-data heterogeneity typically found in web tables and proposed methods to extract data from those tables. Our results show that the architectural components that we designed can: (a) deliver appropriate data that can be used for geospatial data integration, and (b) identify key features to perform integrated data visualization on maps and charts. We also discussed various challenges and proposed ideas to solve them. Our future work is to improve the current architectural system components that we described here and make a more exhaustive evaluation on much larger corpora.

Acknowledgments

We would like to thank Roberto Tamassia, Matteo Palmonari, Tom Theis, Ning Ai, Sajib Derrible, Sam Dorevitch, Naphtali Rische, and Goce Trajceviski for useful discussions.

¹⁴<http://dbpedia.org/>

¹⁵<http://www.opengeospatial.org/standards/geosparql>

¹⁶www.genome.jp/kegg/pathway.html

Thanks are due to Matt Dumford and Anna Anderson for help with the software development. We also acknowledge the constructive input of the anonymous reviewers. This work was supported in part by NSF Awards CCF-1331800, IIS-1213013, IIS-1143926, and IIS-0812258, and by a UIC-IPCE Civic Engagement Research Fund Award.

References

- [1] G. Andogah, G. Bouma, J. Nerbonne, and E. Koster. Place-name Ambiguity Resolution. *LREC Workshop on Methodologies and Resources for Processing Spatial Knowledge*, pages 4–10, 2008.
- [2] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller. Triplify: Light-weight Linked Data Publication from Relational Databases. In *International World Wide Web Conference (WWW)*, pages 621–630. ACM, 2009.
- [3] R. Battle and D. Kolas. Enabling the Geospatial Semantic Web with Parliament and GeoSPARQL. *Semantic Web*, 3(4):355–370, 2012.
- [4] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, pages 29–37, May 2001.
- [5] C. Bizer. D2R MAP – A Database to RDF Mapping Language. In *International World Wide Web Conference (WWW)*, pages 20–24, 2003.
- [6] C. Bizer and R. Cyganiak. D2R Server – Publishing Relational Databases on the Semantic Web. In *International Semantic Web Conference (ISWC)*, page 26, 2006.
- [7] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Webtables: Exploring the Power of Tables on the Web. *PVLDB*, 1(1):538–549, 2008.
- [8] E. Crestan and P. Pantel. Web-Scale Knowledge Extraction from Semi-Structured Tables. In *International World Wide Web Conference (WWW)*, pages 1081–1082. ACM, 2010.
- [9] I. F. Cruz, S. Borisov, M. A. Marks, and T. R. Webb. Measuring Structural Similarity Among Web Documents: Preliminary Results. In *International Conference on Electronic Publishing (EP)*, number 1375 in Lecture Notes in Computer Science, pages 513–524. Springer, 1998.
- [10] I. F. Cruz, V. R. Ganesh, C. Caletti, and P. Reddy. GIVA: A Semantic Framework for Geospatial and Temporal Data Integration, Visualization, and Analytics. In *ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems (ACM GIS)*, 2013.
- [11] I. F. Cruz, F. Palandri Antonelli, and C. Stroe. Agreement-Maker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB*, 2(2):1586–1589, 2009.
- [12] I. F. Cruz, M. Palmonari, F. Caimi, and C. Stroe. Towards “On the Go” Matching of Linked Open Data Ontologies. In *IJCAI Workshop Discovering Meaning On the Go in Large & Heterogeneous Data (LHD)*, pages 37–42, 2011.
- [13] I. F. Cruz, M. Palmonari, F. Caimi, and C. Stroe. Building Linked Ontologies with High Precision Using Subclass Mapping Discovery. *Artificial Intelligence Review*, 40(2):127–145, 2013.
- [14] I. F. Cruz and W. Sunna. Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS, Special Issue on Semantic Similarity Measurement and Geospatial Applications*, 12(6):683–711, 2008.
- [15] I. F. Cruz, W. Sunna, N. Makar, and S. Bathala. A Visual Tool for Ontology Alignment to Enable Geospatial Interoperability. *Journal of Visual Languages and Computing*, 18(3):230–254, 2007.
- [16] G. Dennis Jr, B. T. Sherman, D. A. Hosack, J. Yang, W. Gao, H. C. Lane, and Lempicki. DAVID: Database for Annotation, Visualization, and Integrated Discovery. *Genome Biology*, 4(5):P3, 2003.
- [17] H. Gonzalez, A. Y. Halevy, C. S. Jensen, A. Langen, J. Madhavan, R. Shapley, W. Shen, and J. Goldberg-Kidon. Google Fusion Tables: Web-Centered Data Management and Collaboration. In *ACM SIGMOD International Conference on Management of Data*, pages 1061–1066. ACM, 2010.
- [18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
- [19] B. He, M. Patel, Z. Zhang, and K. C.-C. Chang. Accessing the Deep Web: A Survey. *Communications of the ACM*, 50(5):94–101, 2007.
- [20] K.-S. Kim, K. Zettsu, Y. Kidawara, and Y. Kiyoki. Moving Phenomenon: Aggregation and Analysis of Geotime-Tagged Contents on the Web. In *International Symposium on Web and Wireless Geographical Information Systems (W2GIS)*, pages 7–24. Springer, 2009.
- [21] C. A. Knoblock, P. Szekely, J. L. Ambite, A. Goel, S. Gupta, K. Lerman, M. Muslea, M. Taheriyani, and P. Mallick. Semi-automatically Mapping Structured Sources into the Semantic Web. In *International Semantic Web Conference (ISWC)*, pages 375–390. Springer, 2012.
- [22] J. L. Leidner. Toponym Resolution in Text: “Which Sheffield is it?”. In *International ACM SIGIR Conference (SIGIR)*, page 602. Citeseer, 2004.
- [23] J. L. Leidner, G. Sinclair, and B. Webber. Grounding Spatial Named Entities for Information Extraction and Question Answering. In *HLT-NAACL Workshop on Analysis of Geographic References*, volume 1, pages 31–38. Association for Computational Linguistics, 2003.
- [24] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. Spatio-Textual Spreadsheets: Geotagging via Spatial Coherence. In *ACM Sigspatial International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS)*, pages 524–527. ACM, 2009.
- [25] B. Liu and Y. Zhai. NET—A System for Extracting Web Data from Flat and Nested Data Records. In *International Conference on Web Information Systems Engineering (WISE)*, pages 487–495. Springer, 2005.
- [26] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.
- [27] B. McBride. Jena: A Semantic Web Toolkit. *IEEE Internet Computing*, 6(6):55–59, 2002.
- [28] K. S. McCurley. Geospatial Mapping and Navigation of the Web. In *International World Wide Web Conference (WWW)*, pages 221–229. ACM, 2001.
- [29] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–244, 1990.
- [30] V. Mulwad, T. Finin, and A. Joshi. A Domain Independent Framework for Extracting Linked Semantic Data from Tables. In *Search Computing*, volume 7538 of *Lecture Notes in Computer Science*, pages 16–33. Springer, 2012.
- [31] S. Overell and S. Ruger. Using Co-occurrence Models for Placename Disambiguation. *International Journal of Geographical Information Science*, 22(3):265–287, 2008.
- [32] G. Quercini and C. Reynaud. Entity Discovery and Annotation in Tables. In *International Conference on Extending Database Technology (EDBT)*, pages 693–704. ACM, 2013.
- [33] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [34] H. Rijgersberg, M. van Assem, and J. Top. Ontology of Units of Measure and Related Concepts. *Semantic Web*, 4(1):3–13, 2013.
- [35] R. Rowlett. How Many? A Dictionary of Units of Measurement, 2000. <http://www.unc.edu/~rowlett/units>.
- [36] Z. Syed, T. Finin, V. Mulwad, and A. Joshi. Exploiting a Web of Semantic Data for Interpreting Tables. In *Web Science Conference*, 2010.
- [37] W. R. Tobler. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, 46:234–240, 1970.
- [38] P. Venetis, A. Halevy, J. Madhavan, M. Pa¸ca, W. Shen, F. Wu, G. Miao, and C. Wu. Recovering Semantics of Tables on the Web. *PVLDB*, 4(9):528–538, 2011.