

# Deep Inside the Tables: Semantic Expressiveness of Semi-Structured Data

BY

VENKAT RAGHAVAN GANESH SEKAR

B.Tech., SASTRA University, 2009

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Computer Science  
in the Graduate College of the  
University of Illinois at Chicago, 2014

Chicago, Illinois

Defense Committee:

Isabel Cruz, Chair and Advisor

Brian Ziebart, University of Illinois at Chicago

Matteo Palmonari, University of Milan-Bicocca

Dedicated to my parents,

Prema & Ganesh Sekar

## ACKNOWLEDGMENTS

First of all, I would like to thank my research advisor Dr. Isabel Cruz. I am very happy to have been assisting in her research throughout my stay at UIC. She gave an overwhelming support without which I wouldn't have come this far in my research. Needless to say, it was her invaluable suggestions and advises that inculcated several attributes of a researcher. I would also like to thank her for providing me several wonderful opportunities, and for giving me the freedom to initiate and discuss any new ideas that came along the way. Almost every meeting with Dr. Cruz has been a learning experience. I must also thank her for providing a proper infrastructure to ADVIS lab and for inspecting them on a timely basis that helped in smooth conduct of the research.

My sincere thanks to all of my friends at ADVIS Lab (Claudio Caletti, Cosmin Stroe, Francesco Loperete, Iman Mirzaei, Matthew Dumford and Pavan Kumar) for their assistance on various events, for spending their time on interesting discussions, and for those sleepless nights working together for various submissions. I would specially like to thank Iman for his work on spatial disambiguation matcher and Claudio for defining the hierarchy of spatial data formats. I would also like to thank the CS department staffs for their indirect support.

Last but not the least, I would like to thank my family for their patience and support.

VRGS

## TABLE OF CONTENTS

| <u>CHAPTER</u>                                  | <u>PAGE</u> |
|---|-------------|
| <b>1 INTRODUCTION . . . . .</b>                 | <b>1</b>    |
| <b>2 SEMI-STRUCTURED DATA . . . . .</b>         | <b>5</b>    |
| 2.1 Web Table . . . . .                         | 6           |
| 2.2 Spreadsheet . . . . .                       | 8           |
| 2.3 Heterogeneity . . . . .                     | 8           |
| 2.3.1 Structural . . . . .                      | 9           |
| 2.3.2 Conceptual . . . . .                      | 10          |
| 2.3.3 Metadata . . . . .                        | 10          |
| <b>3 TABLE REPRESENTATION . . . . .</b>         | <b>11</b>   |
| 3.1 Header rows . . . . .                       | 11          |
| 3.2 Data rows . . . . .                         | 11          |
| 3.3 Model . . . . .                             | 12          |
| <b>4 TABLE ANNOTATION . . . . .</b>             | <b>14</b>   |
| 4.1 Natural Language Processing . . . . .       | 14          |
| 4.2 Named Entity Recognition . . . . .          | 15          |
| 4.3 Annotation Pipeline . . . . .               | 17          |
| 4.3.1 Preprocessing . . . . .                   | 18          |
| 4.3.2 Feature Taggers . . . . .                 | 19          |
| 4.3.2.1 Concept . . . . .                       | 19          |
| 4.3.2.2 Modifier . . . . .                      | 20          |
| 4.3.2.3 Measurement Units and Symbols . . . . . | 21          |
| 4.3.2.4 Uncertainty . . . . .                   | 23          |
| 4.3.2.5 Spatial and Temporal . . . . .          | 23          |
| 4.3.3 Spatial . . . . .                         | 23          |
| 4.3.3.1 Spatial Identification. . . . .         | 24          |
| 4.3.3.2 Geocoding. . . . .                      | 25          |
| 4.3.4 Temporal . . . . .                        | 28          |
| 4.4 Annotation Profile . . . . .                | 28          |
| <b>5 CAPTION ANNOTATION . . . . .</b>           | <b>30</b>   |
| 5.1 Characteristics . . . . .                   | 30          |
| 5.2 Preprocessing . . . . .                     | 31          |
| 5.2.1 Title case . . . . .                      | 32          |
| 5.2.2 Case refinement . . . . .                 | 32          |

## TABLE OF CONTENTS (Continued)

| <u>CHAPTER</u> |   | <u>PAGE</u> |
|----------------|---|-------------|
|                | 5.3 Caption Annotation . . . . .                | 33          |
| <b>6</b>       | <b>SEMANTIC GRAPH . . . . .</b>                 | <b>34</b>   |
|                | 6.1 Semantic Graph . . . . .                    | 34          |
|                | 6.1.1 Triple Representation . . . . .           | 35          |
|                | 6.1.2 Construction . . . . .                    | 36          |
|                | 6.1.2.1 Table Graph . . . . .                   | 36          |
|                | 6.1.2.2 Table Caption . . . . .                 | 38          |
|                | 6.2 Example . . . . .                           | 41          |
| <b>7</b>       | <b>APPLICATIONS . . . . .</b>                   | <b>44</b>   |
|                | 7.1 Data Cube . . . . .                         | 44          |
|                | 7.2 GIVA . . . . .                              | 47          |
|                | 7.2.1 Data Extraction . . . . .                 | 48          |
|                | 7.2.2 Data Translation . . . . .                | 49          |
|                | 7.2.3 Ontology Extraction . . . . .             | 49          |
|                | 7.2.4 Matching . . . . .                        | 49          |
|                | 7.2.4.1 Semantic Matching . . . . .             | 50          |
|                | 7.2.4.2 Spatio-Temporal Matching . . . . .      | 50          |
|                | 7.2.5 Storage Systems and Application . . . . . | 51          |
| <b>8</b>       | <b>EXPERIMENTS . . . . .</b>                    | <b>53</b>   |
|                | 8.1 Datasets . . . . .                          | 53          |
|                | 8.1.1 Complex Tables . . . . .                  | 53          |
|                | 8.1.1.1 Decision Tree . . . . .                 | 54          |
|                | 8.1.2 Wikipedia tables . . . . .                | 54          |
|                | 8.1.3 Captions . . . . .                        | 56          |
|                | 8.1.3.1 Google Tables . . . . .                 | 56          |
|                | 8.2 Evaluation . . . . .                        | 57          |
|                | 8.2.1 Table Annotation . . . . .                | 58          |
|                | 8.2.2 Caption Annotation . . . . .              | 60          |
|                | 8.2.3 Semantic Graph . . . . .                  | 61          |
| <b>9</b>       | <b>RELATED WORK . . . . .</b>                   | <b>63</b>   |
| <b>10</b>      | <b>FUTURE WORK . . . . .</b>                    | <b>65</b>   |
| <b>11</b>      | <b>CONCLUSIONS . . . . .</b>                    | <b>68</b>   |
|                | <b>CITED LITERATURE . . . . .</b>               | <b>68</b>   |
|                | <b>APPENDICES . . . . .</b>                     | <b>74</b>   |

## TABLE OF CONTENTS (Continued)

| <u>CHAPTER</u>       | <u>PAGE</u> |
|----------------------|-------------|
| Appendix A . . . . . | 75          |
| Appendix B . . . . . | 77          |
| VITA . . . . .       | 79          |

## LIST OF TABLES

| <u>TABLE</u> |   | <u>PAGE</u> |
|--------------|---|-------------|
| I            | USAGE OF <i>COLSPAN</i> AND <i>ROWSPAN</i> IN A HTML TABLE . .            | 7           |
| II           | WIDELY OCCURRING POS TAGS IN TABULAR DATA . . . . .                       | 16          |
| III          | TABLE A: WATER INFORMATION RECORDED AT WATER STATIONS IN THE U.S. . . . . | 18          |
| IV           | SINGLE CONCEPT FOR THE ENTIRE TABLE . . . . .                             | 20          |
| V            | ADVERBS AS A MODIFIER TO THE CONCEPT . . . . .                            | 21          |
| VI           | SYMBOLIC REPRESENTATIONS AND THEIR DESCRIPTIONS                           | 22          |
| VII          | TABLE A: WATER LEVELS MEASURED AT WATER STATIONS IN ILLINOIS . . . . .    | 30          |
| VIII         | EFFECT OF TITLE CASE IN TABLE CAPTION . . . . .                           | 31          |
| IX           | TABLE GRAPH - LINKAGE . . . . .   | 37          |
| X            | TABLE CAPTION - LINKAGE . . . . .   | 39          |
| XI           | EXAMPLE - TABLE CAPTION LINKAGE . . . . .                                 | 40          |
| XII          | TABLE CAPTION - HIERARCHY . . . . .                                       | 40          |
| XIII         | DECISION TREE FEATURES FOR TABLE EXTRACTION. . . .                        | 55          |
| XIV          | KEYWORDS FOR TABLE SEARCH. . . . .  | 57          |
| XV           | FEATURE DISTRIBUTION IN DATASETS. . . . .                                 | 58          |
| XVI          | PERCENTAGE ACCURACY OF TABLE ANNOTATION. . . . .                          | 59          |
| XVII         | PERCENTAGE ACCURACY OF CAPTION ANNOTATION. . . .                          | 61          |

## LIST OF FIGURES

| <b><u>FIGURE</u></b> |   | <b><u>PAGE</u></b> |
|----------------------|---|--------------------|
| 1                    | Semi-structured data in a HTML table. . . . .                         | 3                  |
| 2                    | Web page containing a complex table. . . . .                          | 6                  |
| 3                    | A spreadsheet showing poverty statistics. . . . .                     | 9                  |
| 4                    | Table Representation for the complex web table shown in Figure 1. . . | 12                 |
| 5                    | Table annotation pipeline . . . . .                                   | 17                 |
| 6                    | Disambiguation in Geocoding . . . . .                                 | 24                 |
| 7                    | Annotation Profile . . . . .  | 29                 |
| 8                    | A simple ontology with <i>hasA</i> relationships . . . . .            | 35                 |
| 9                    | Triples as a graph . . . . .  | 36                 |
| 10                   | Structural Linkage . . . . .  | 38                 |
| 11                   | Table Caption - Semantic Graph . . . . .                              | 42                 |
| 12                   | Complete Semantic Graph . . . . .                                     | 43                 |
| 13                   | Caption for LOF . . . . .   | 46                 |
| 14                   | GIVA framework . . . . .  | 47                 |
| 15                   | Hierarchy of spatial data types. . . . .                              | 48                 |
| 16                   | GIVA Application. . . . .   | 52                 |
| 17                   | Annotation graph. . . . .   | 62                 |
| 18                   | Web Table with relational data. . . . .                               | 64                 |



## LIST OF ABBREVIATIONS

|        |  |
|--------|--|
| GIS    | Geospatial Information Systems   |
| GDAL   | Geospatial Data Abstraction Library                                    |
| RDF    | Resource Description Framework   |
| OWL    | Web Ontology Language  |
| GIVA   | Geospatial and temporal data Integration, Visualization and Analytics. |
| SPARQL | SPARQL Protocol And RDF Query Language                                 |
| NLP    | Natural Language Processing  |
| NER    | Named Entity Recognition   |
| HTML   | HyperText Markup Language  |
| XML    | Extensible Markup Language   |
| OLAP   | On-Line Analytical Processing  |
| POS    | Parts-Of-Speech  |
| SDMX   | Statistical Data and Metadata eXchange                                 |

## SUMMARY

The web contains huge amount of semi-structured data in the form of tables and spreadsheets that are pertinent for various statistical data analysis or visualization. Manual processing of these tabular data is tedious because of their heterogeneity in structure, concept and metadata. Further, much of the information present in them do not have explicit metadata introducing difficulties in understanding the table semantics which is critical to automatically process these data and to leverage the data integration process. In this thesis, we (a) study in-depth about semi-structured (tabular) data on the web; (b) discuss the complexities in processing them; (c) propose automatic methods to abstract their semantics by annotating various features inside the tables; (d) introduce algorithms to construct a semantic graph by resolving different levels of heterogeneities. We evaluate our approach on a set of highly complex tables retrieved from different domains and also discuss about the impact of our work in practical scenarios and in the field of Semantic Web.

# CHAPTER 1

## INTRODUCTION

Web tables are considered to be one of the most prominent form that contain semi-structured data mainly due to its huge availability on the web. As of 2008, there were 14.1 billion HTML tables out of which 154 million contained high quality relational data [1]. Besides these tables, we also notice that several government organizations share a diverse range of reports and statistics in the form of spreadsheets, tables in PDF documents, XMLs and CSVs that add up further to the quantity of useful semi-structured data on the web. Few examples include census report from U.S. Census Bureau<sup>1</sup>, weather and water flow related information from National Oceanic and Atmospheric Administration (NOAA)<sup>2</sup>, and several locally administered websites for large cities such as Chicago Data Portal (CDP)<sup>3</sup> and New York City Open Data<sup>4</sup>. Some of these web sites provide specific ways to query and visualize their data independently. For instance, CDP provides facilities to visualize certain categories of data on map and to add multiple map layers to create an overlay effect for analysis. This is a major obstacle for any interdisciplinary research that demands the interactions between the data available in various

---

<sup>1</sup><https://www.census.gov/>

<sup>2</sup><http://www.noaa.gov/ocean.html>

<sup>3</sup><https://data.cityofchicago.org/>

<sup>4</sup><https://nycopendata.socrata.com/>

heterogeneous domains and sources for data analysis. For instance, a public health scientist would be interested in data from Census Bureau and also the National Weather Service (NWS) for, say, analyzing the effect of weather on a subset of population in an area. Thus, a platform for data integration has to be laid out. But, the performance of data integration depends on the way the data will be extracted and represented.

In most cases, the data present within these tables may be a mere integration of several (possibly) structured data sources [2] with a motive to make them understandable by the humans. For example, the web table shown in Figure 1 could be built by extracting data from structured sources on *Water*, *Precipitation* and *River*. In some scenarios such as government reports, spreadsheet applications are used for data entry which vary in structures. Because the conversion of spreadsheets to HTML tables is simple, the same level of heterogeneity gets transferred to the web.

Firstly, manual extraction of such data is time-intensive and becomes nearly impossible when dealing with large number of heterogeneous data. Current automatic techniques only perform *blind* data extraction that does not understand the content and that can only be applied on tables that have clear headers and data similar to that of a relational table in RDBMS. However, we notice that this is not sufficient. For instance, it is necessary to identify the semantics of a column header like *Average rainfall (mm) from 2010-2013* as a header that (1) contain data about *Rainfall*; (2) the data in that column are *Average* amounts; (3) the units of measurement is *millimeters*; (4) the data is valid or applicable only for the temporal range *2010 to 2013*. Apart from this, there is the problem of representing the extracted data which relies extensively

**Table A: Water information recorded at water stations in the U.S.**

Excel | CSV

|                          | Water Temperature              |           |         | Average Precipitation | Maximum water level |
|--------------------------|--------------------------------|-----------|---------|-----------------------|---------------------|
|                          | Recent (F)                     | Oct 16-31 | Nov (F) | Mar-Nov (in)          |                     |
| Reedy Point, DE          | 34.3<br>(01/31/2014 20:54 UTC) | 59 F      | 52      | 38.6                  | 12 ft               |
| Annapolis, MD            | 30.7<br>(01/31/2014 20:54 UTC) | 60 F      | --      | 33.65                 | 14                  |
| La Grange                | N/A                            | 58        | 57      |                       | 13.2 ft             |
| Tacony-Palmyra Bridgy NJ | 32.1 (01/31/2014 20:48 UTC)    | 62        | 57      | 42.6                  | 12.3                |

Figure 1. Semi-structured data in a HTML table.

on the original structure. We see that most of these data have complex and fuzzy schema with variety of information embedded in it. The table shown in Figure 1 with nested headers and heterogeneous metadata illustrates these issues. Thus, in order to perform a meaningful search over these tables and to integrate them, special techniques needs to be developed.

In this thesis, we will study intensively on the semi-structured data present in tables and propose automatic methods to extract and represent them in a format that retains the semantics such that no tiny information is lost.

The rest of this thesis is organized as follows: Chapter 2 introduces semi-structured data and their heterogeneity. Chapter 3 describes the preliminary data structure to represent a complex table. Chapter 4 and 5 explains the several annotation components and introduces an

algorithm to develop an *Annotation Profile*. Chapter 6 introduces an algorithm to create the semantic graph and also discusses possible applications in real-life scenario. Chapter 8 describes the experimental setup, creation of test data and experimental results of various components. Chapter 9 discusses the previous work on extraction. We conclude and discuss the future work in Chapters 11 and 10 respectively.

## CHAPTER 2

### SEMI-STRUCTURED DATA

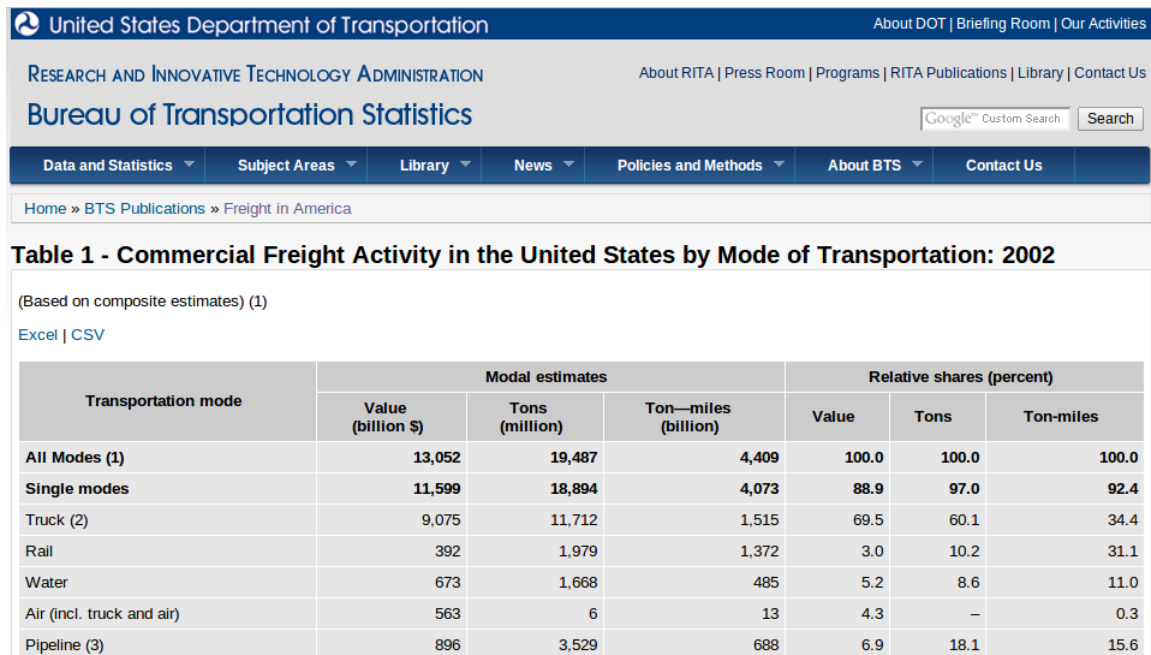
The electronic data on the web can be broadly classified into three main categories having specific characteristics:

**Unstructured data.** Data that has no defined schema is called unstructured data. They are mostly raw text data which may, sometimes, include certain objects such as images and videos.

**Structured data.** Data that conforms to a well defined schema and other data descriptions is generally referred to as structured data. A well known example is the table in a relational database.

**Semi-Structured data.** Data that combines the characteristics of structured and unstructured is called semi-structured data. They are neither structured nor entirely unstructured. The data representation is thus very flexible such that no single schema or structure could represent the entire data. Examples include the data created using HTML or XML tags, and BibTex file.

Several main aspects and various forms of semi-structured data are explained by Abiteboul [2]. In this section, we discuss only about the data that are present in web tables and spreadsheets.



United States Department of Transportation

RESEARCH AND INNOVATIVE TECHNOLOGY ADMINISTRATION

Bureau of Transportation Statistics

Table 1 - Commercial Freight Activity in the United States by Mode of Transportation: 2002

(Based on composite estimates) (1)

Excel | CSV

| Transportation mode       | Modal estimates       |                   |                        | Relative shares (percent) |       |           |
|---------------------------|-----------------------|-------------------|------------------------|---------------------------|-------|-----------|
|                           | Value<br>(billion \$) | Tons<br>(million) | Ton—miles<br>(billion) | Value                     | Tons  | Ton-miles |
| All Modes (1)             | 13,052                | 19,487            | 4,409                  | 100.0                     | 100.0 | 100.0     |
| Single modes              | 11,599                | 18,894            | 4,073                  | 88.9                      | 97.0  | 92.4      |
| Truck (2)                 | 9,075                 | 11,712            | 1,515                  | 69.5                      | 60.1  | 34.4      |
| Rail                      | 392                   | 1,979             | 1,372                  | 3.0                       | 10.2  | 31.1      |
| Water                     | 673                   | 1,668             | 485                    | 5.2                       | 8.6   | 11.0      |
| Air (incl. truck and air) | 563                   | 6                 | 13                     | 4.3                       | —     | 0.3       |
| Pipeline (3)              | 896                   | 3,529             | 688                    | 6.9                       | 18.1  | 15.6      |

Figure 2. Web page containing a complex table.

## 2.1 Web Table

Web Tables are constructed for several purposes mainly because of its ability to structure the information in many different ways such as HTML forms, Calendar, relational data, reports and many more. Hyper Text Markup Language (HTML) is a universally understood language to the web browsers to publish information over the World Wide Web [3]. It contains a set of markup tags that are used to create an HTML document (herein referred to as a *web page*). A sample template of a web page containing a table shown in Table I and brief description of some important markup tags relevant to this research are given below:



TABLE I  
USAGE OF *COLSPAN* AND *ROWSPAN* IN A HTML TABLE

| Location    | Water level (ft) |     |
|-------------|------------------|-----|
|             | Avg              | Max |
| Des Plaines | 12.5             | 30  |

```

1 | <html>
2 |   <head>
3 |     <title> This is a title </title>
4 |     <meta name='description' content='Thesis'>
5 |     <style> table, th {color:red;} </style>
6 |     <script src='test.js'>
7 |   </head>
8 |   <body>
9 |     <table>
10 |       <tr>
11 |         <th rowspan='2'>Location</th>
12 |         <th colspan='2'>Water level (ft)</th>
13 |       </tr>
14 |       <tr>
15 |         <td>Avg</td>
16 |         <td>Max</td>
17 |       </tr>
18 |       <tr>
19 |         <td>Des Plaines</td>
20 |         <td>12.5</td>
21 |         <td>30</td>
22 |       </tr>
23 |     </table>
24 |   </body>
25 | </html>

```

The `<head>` block contains markup tags that provide metadata and other basic resources to the main document such as `<title>` for web page title, `<meta>` to provide the metadata, `<style>` to apply style information (sometimes `<link>` is used to attach a Cascaded Style Sheet),

and `<script>` to apply a JavaScript or a VBScript. The `<body>` block contains markup tags that help in the representation of information on the web page as shown in Figure 2. The tables constructed using HTML markup tags are referred to as **Web Tables**. A few important HTML tags used to construct a table are: (1) `<table>` as a wrapper tag; (2) `<tr>` to define a row; (3) `<th>` to define a table header for a column; (4) `<td>` to define a column. Besides these tags, there are (a) positional attributes such as *width*, *height* and *valign*; (b) cell merging attributes such as *colspan* and *rowspan*; (c) style attributes such as *style*, *bgcolor*, *border*, *cellpadding* and *cell spacing*. The flexibility of using the combination of these tags and attributes can make the table very complex yet visually pleasing to human eyes. Thus, these tables are generally said to have a self-describing structure and metadata.

## 2.2 Spreadsheet

Spreadsheets are worksheets with rows and columns that are created using softwares such as Microsoft<sup>TM</sup> Excel<sup>®</sup> or OpenOffice.org<sup>TM</sup>. Unlike the HTML tags, these softwares provide the functionality to apply mathematical functions and to design the table through a GUI. This assists in easy report generation. Some of these tools can even convert such reports in to a web table for publishing over the web. Majority of government organizations and corporate companies use spreadsheets to create different reports and other forms of statistical data. See Figure 3. In the field of On-Line Analytical Processing (OLAP), these types of data are referred to as multi-dimensional data. [4]

## 2.3 Heterogeneity

We categorize the heterogeneity of these semi-structured data (tables) as described below:

|    | A        | B       | C      | D    | E       | F      | G    | H      | I     | J    | K      | L     | M    | N      | O     | P    |
|----|----------|---------|--------|------|---------|--------|------|--------|-------|------|--------|-------|------|--------|-------|------|
| 1  |          |         |        |      |         |        |      |        |       |      |        |       |      |        |       |      |
| 2  |          |         |        |      |         |        |      |        |       |      |        |       |      |        |       |      |
| 3  |          |         |        |      |         |        |      |        |       |      |        |       |      |        |       |      |
| 4  |          |         |        |      |         |        |      |        |       |      |        |       |      |        |       |      |
| 5  |          |         |        |      |         |        |      |        |       |      |        |       |      |        |       |      |
| 6  |          |         |        |      |         |        |      |        |       |      |        |       |      |        |       |      |
| 7  | 2012     | 310,648 | 46,496 | 15.0 | 270,570 | 38,803 | 14.3 | 40,078 | 7,693 | 19.2 | 18,193 | 2,252 | 12.4 | 21,885 | 5,441 | 24.9 |
| 8  | 2011     | 308,456 | 46,247 | 15.0 | 268,490 | 38,661 | 14.4 | 39,966 | 7,586 | 19.0 | 17,934 | 2,233 | 12.5 | 22,032 | 5,353 | 24.3 |
| 9  | 2010 17/ | 306,130 | 46,343 | 15.1 | 266,723 | 38,485 | 14.4 | 39,407 | 7,858 | 19.9 | 17,344 | 1,954 | 11.3 | 22,063 | 5,904 | 26.8 |
| 10 | 2010     | 305,688 | 46,180 | 15.1 | 267,487 | 38,568 | 14.4 | 38,201 | 7,611 | 19.9 | 16,797 | 1,906 | 11.3 | 21,403 | 5,706 | 26.7 |
| 11 | 2009     | 303,820 | 43,569 | 14.3 | 266,223 | 36,407 | 13.7 | 37,597 | 7,162 | 19.0 | 16,024 | 1,736 | 10.8 | 21,573 | 5,425 | 25.1 |
| 12 | 2008     | 301,041 | 39,829 | 13.2 | 264,314 | 33,293 | 12.6 | 36,727 | 6,536 | 17.8 | 15,470 | 1,577 | 10.2 | 21,257 | 4,959 | 23.3 |
| 13 | 2007     | 298,699 | 37,276 | 12.5 | 261,456 | 31,126 | 11.9 | 37,243 | 6,150 | 16.5 | 15,050 | 1,426 | 9.5  | 22,193 | 4,724 | 21.3 |
| 14 | 2006     | 296,450 | 36,460 | 12.3 | 259,199 | 30,790 | 11.9 | 37,251 | 5,670 | 15.2 | 14,534 | 1,345 | 9.3  | 22,716 | 4,324 | 19.0 |
| 15 | 2005     | 293,135 | 36,950 | 12.6 | 257,513 | 31,080 | 12.1 | 35,621 | 5,870 | 16.5 | 13,881 | 1,441 | 10.4 | 21,740 | 4,429 | 20.4 |
| 16 | 2004 14/ | 290,617 | 37,040 | 12.7 | 255,443 | 31,023 | 12.1 | 35,173 | 6,017 | 17.1 | 13,505 | 1,326 | 9.8  | 21,669 | 4,691 | 21.6 |
| 17 | 2003     | 287,699 | 35,861 | 12.5 | 253,478 | 29,965 | 11.8 | 34,221 | 5,897 | 17.2 | 13,128 | 1,309 | 10.0 | 21,094 | 4,588 | 21.7 |
| 18 | 2002     | 285,317 | 34,570 | 12.1 | 251,881 | 29,012 | 11.5 | 33,437 | 5,558 | 16.6 | 12,832 | 1,285 | 10.0 | 20,605 | 4,273 | 20.7 |
| 19 | 2001     | 281,475 | 32,907 | 11.7 | 249,053 | 27,698 | 11.1 | 32,422 | 5,209 | 16.1 | 11,962 | 1,186 | 9.9  | 20,460 | 4,023 | 19.7 |
| 20 | 2000 12/ | 278,944 | 31,581 | 11.3 | 247,162 | 26,680 | 10.8 | 31,782 | 4,901 | 15.4 | 11,785 | 1,060 | 9.0  | 19,997 | 3,841 | 19.2 |
| 21 | 1999 11/ | 276,208 | 32,791 | 11.9 | 246,256 | 27,757 | 11.3 | 29,952 | 5,034 | 16.8 | 11,065 | 996   | 9.0  | 18,886 | 4,039 | 21.4 |
| 22 | 1998     | 271,059 | 34,476 | 12.7 | 244,636 | 29,707 | 12.1 | 26,424 | 4,769 | 18.0 | 9,864  | 1,087 | 11.0 | 16,560 | 3,682 | 22.2 |
| 23 | 1997     | 268,480 | 35,574 | 13.3 | 242,219 | 30,336 | 12.5 | 26,261 | 5,238 | 19.9 | 9,732  | 1,111 | 11.4 | 16,529 | 4,127 | 25.0 |
| 24 | 1996     | 266,218 | 36,529 | 13.7 | 240,459 | 31,117 | 12.9 | 25,759 | 5,412 | 21.0 | 9,043  | 936   | 10.3 | 16,716 | 4,476 | 26.8 |
| 25 | 1995     | 263,733 | 36,425 | 13.8 | 239,206 | 30,972 | 12.9 | 24,527 | 5,452 | 22.2 | 7,904  | 833   | 10.5 | 16,623 | 4,619 | 27.8 |
| 26 | 1994     | 261,616 | 38,059 | 14.5 | 238,650 | 32,865 | 13.8 | 22,967 | 5,194 | 22.6 | 7,097  | 668   | 9.4  | 15,869 | 4,526 | 28.5 |
| 27 | 1993 10/ | 259,278 | 39,265 | 15.1 | 236,745 | 34,086 | 14.4 | 22,533 | 5,179 | 23.0 | 6,973  | 707   | 10.1 | 15,560 | 4,472 | 28.7 |

Figure 3. A spreadsheet showing poverty statistics.

### 2.3.1 Structural

Structural heterogeneity arise due to the merging of cells. This process is technically referred to as *cell spanning*. For instance, we can clearly note this in Table I where *Location* spans across two rows and *Water level (ft)* spans across two columns. We refer to these differences as structural heterogeneities. From a high-level view, this may be intuitively considered as a simple hierarchical structure. However, we find that it is not the case for several reasons which we explain in Chapter 6. Tables that do not have any such cell spanning and those that contain clean header structure (single row) can be compared to that of a structure maintained in a relational table.

### 2.3.2 Conceptual

One of the important characteristics of semi-structured data is that their data elements are eclectic [2]. This introduces the heterogeneity in concepts present in them. For instance, a table describing the climate of different U.S. cities may have information about different concepts such as *Temperature*, *Rainfall* and *Snowfall*. The generation of such data could be the result of database queries executed on three different relational tables.

### 2.3.3 Metadata

Metadata is the data that describes data. In tables, it could be the *table caption* that summarizes the table or it can be some implicit information present within the table headers or data rows. For example, following our previous example from Section 2.3.2, the table on climate may contain headers such as *Average temperature (F)*, *Maximum rainfall (mm)* and *Average snowfall (in)*. If we look at the first header, we note that there are different *features* in it such as *Average* (modifier), *Temperature* (concept) and *F* referring to Fahrenheit (unit). These features constitute the metadata. The identification of this metadata is one of the important process in understanding the data. Chapter 4 and Chapter 5 further describes this heterogeneity and introduces methods to identify these metadata.

## CHAPTER 3

### TABLE REPRESENTATION

Tables have a grid structure with rows and columns. Each individual unit is referred to as a *cell*. Within a table, there are two categories of rows as described in the following sections:

#### 3.1 Header rows

Table headers (created generally using `<th>` tag) contain a schema that may describe the table. They are usually present in the first row (in a simple table) or first two to three rows (in a complex table where row spanning occurs). These rows are called *header rows*. The cells present within these rows are referred to as *header cells*. In a table that is aligned horizontally (called as *horizontal tables*), headers may be present in the first column and termed as *header columns*. However, in this thesis we only focus on vertically structured tables.

#### 3.2 Data rows

These rows appear after the *header rows* and contain data that belong to the appropriate header cells. However, this distinction between *header rows* and *data rows* is not always explicit because of improper web table construction (not using `<th>` tags) or unclear layouts (in spreadsheet applications). There have been many techniques presented to handle this issue of separating the schema (*header cells*) and the data (*data cells*) [5, 6, 7]. However, these methods work under an assumption that schema is present only in header cells (which is not the case in many scenarios). For instance, as shown in Figure 1, the *data* cells may contain schema related information such as units or even a timestamp. This demands an alternate model that would

allow the modification of a data cell (or a *feature* of a data cell) to a header cell in such a way it gets attributed to the main schema.

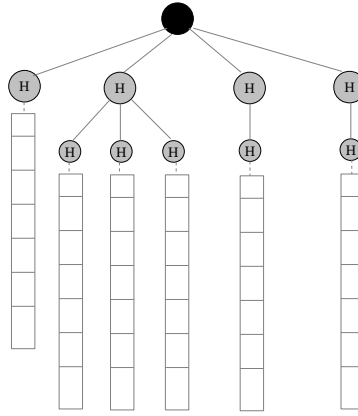


Figure 4. Table Representation for the complex web table shown in Figure 1.

### 3.3 Model

Although several representations are possible for various kinds of table [8, 6], in this thesis, we mainly focus on simple tables and tables with nested header structures. We use a special type of tree  $\mathcal{T}_s$  with number of nodes  $n$  equivalent to the number of initial *header cells*. See Figure 4. We say them ‘initial’ as the tree may change at a later point of time which is described in Chapter 6. The leaf nodes of this tree has a list  $\mathcal{L}_i$  where  $i = 1, 2, 3 \dots k$  ( $k$  being the number of columns in the *data rows*). This representation allows easy transformation between a tree and a matrix whenever required. This is because each node  $n$  in this tree has the ability to

convert the *data rows* into a matrix  $\mathcal{M}$ . For instance, in Figure 4, the second node that has three child nodes contain three independent lists. These lists can be merged to form a matrix  $\mathcal{M}_2$ . This is useful in certain cases which requires merging of cells such as combining *apartment number*, *street name* and *city* to perform Geocoding [9] to identify their geographic coordinates.

## CHAPTER 4

### TABLE ANNOTATION

In order to automatically identify various *features* of the table, we develop an annotation pipeline with different components that apply Natural Language Processing (NLP) and Named Entity Recognition (NER) techniques. The performance of this process is highly critical to semantically model the data. In this chapter, we first provide a brief overview of NLP and NER following which we explain in-depth the different *features* of a table that are available for extraction and also discuss our annotation algorithm.

#### 4.1 Natural Language Processing

Natural Language Processing (NLP) has been proved critical to various research applications involving unstructured textual data that come under the category of Open Information Extraction (Open IE) [11, 12, 13, 14]. Given a well phrased English sentence, NLP parsers such as Stanford Core NLP<sup>1</sup> or Apache OpenNLP<sup>2</sup> provide complete grammatical analysis for them. These analysis may be in the form of Part-of-Speech (POS) information or typed dependency tree [15]. The performance of these parsers depend on “models” that are trained over a high quality text corpus such as news articles and Wikipedia content. Unfortunately, this makes

---

**Acknowledgement:** The work in this chapter is an improved version of a paper presented elsewhere [10].

<sup>1</sup><http://nlp.stanford.edu/software/corenlp.shtml>

<sup>2</sup><https://opennlp.apache.org/>



them work detrimentally on badly phrased sentences such as tweets that require special parser such as CMU Tweet NLP<sup>1</sup>. In both of the cases, the availability of homogeneous data format (grammatical text as in natural language corpus or tweets as in Twitter) makes it possible to build a training model which is often not the case with semi-structured data that is heterogeneous in different ways as discussed in previous sections. However, we take advantage of these parsers with different pre-processing and post-processing steps to use them on a semi-structured data. After analysing large number of tabular data (See Chapter 8), we created a subset of POS (out of 48 tags based on Penn Tree bank [16] tagset) that occur widely in them. These are listed in Table II along with their descriptions.

## 4.2 Named Entity Recognition

Named Entity Recognition (NER) is the process of identifying (annotating) entities automatically such as a name of a person, an organization or other custom entities from unstructured text data. Named entity recognition is an important process that works in parallel with NLP techniques in the field of Information Extraction (IE) [11, 17, 18]. Research on NER is well established with state-of-the-art techniques producing near-human accuracies in entity recognition [19]. Based on the nature of the text content, techniques that use maximum entropy [20], conditional random fields [21] or gazetteer [18] are used to annotate text. Ambiguities are internally handled by these methods that involve the processing of *context* information that are available in the text content. In tabular data, for instance, a cell with the content

---

<sup>1</sup><http://www.ark.cs.cmu.edu/TweetNLP/>

TABLE II  
WIDELY OCCURRING POS TAGS IN TABULAR DATA

| POS Tag | Description                              |
|---------|--|
| NN      | Noun (singular)                          |
| NNS     | Noun (plural)                            |
| NNP     | Proper noun (singular)                   |
| NNPS    | Proper noun (plural)                     |
| RB      | Adverb                                   |
| RBR     | Adverb (comparative)                     |
| RBS     | Adverb (superlative)                     |
| JJ      | Adjective                                |
| JJR     | Adjective (comparative)                  |
| JJS     | Adjective (superlative)                  |
| TO      | to                                       |
| IN      | Preposition or subordinating conjunction |
| VB*     | All forms of verbs                       |

*Washington* may be recognized as a *Location* and also a *Person*. While the context information would be available in an unstructured text such as *Washington is the 13<sup>th</sup> populous state in United States.*, the same context may be present in data columns, table caption or even in other metadata that may be present outside a table. Obtaining this information is highly challenging especially for the tables that contain statistical data. To apply NER on tables, we combine traditional NER (Stanford NER<sup>1</sup>) with custom dictionaries to annotate entities within the tables.

---

<sup>1</sup><http://nlp.stanford.edu/software/CRF-NER.shtml>

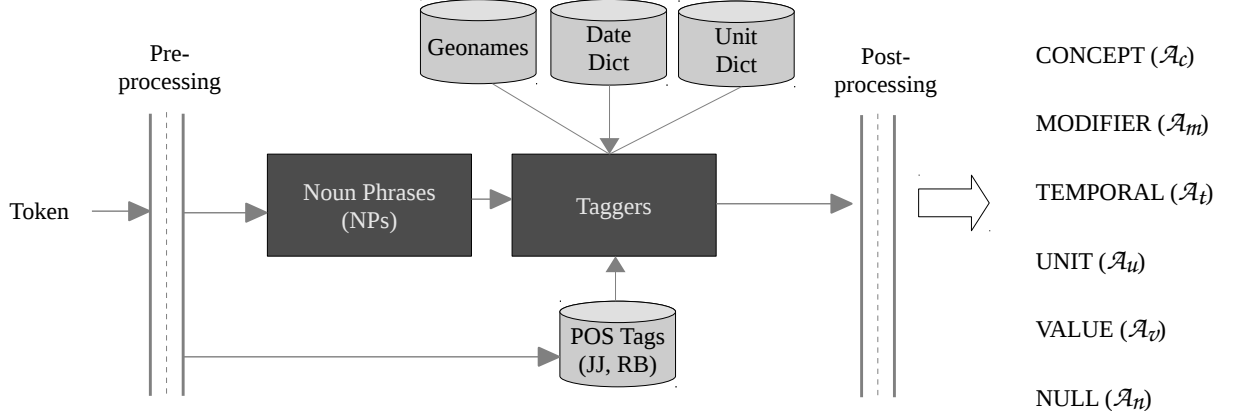


Figure 5. Table annotation pipeline

### 4.3 Annotation Pipeline

In this section, we describe different components of annotation pipeline (See Figure 5) that can automatically identify several features by leveraging NLP and NER techniques. A *feature* is defined as the smallest information unit that is present within the cells. Because of the heterogeneity (as discussed in Chapter 2), one cell may contain more than one feature creating a fuzzy metadata. The annotation pipeline consists of a *pre-processing unit*, a set of *feature taggers* and a *post-processing unit*. We will use our running example shown in Table III throughout this section to explain different components. In the following subsections, we describe various *features*, provide a methodological overview to identify them and introduce our annotation algorithm.

TABLE III

TABLE A: WATER INFORMATION RECORDED AT WATER STATIONS IN THE U.S.

|                           | Water Temperature              |                 |               | Average Precipitation | Maximum water level |
|---------------------------|--------------------------------|-----------------|---------------|-----------------------|---------------------|
|                           | Recent (F)                     | Oct 16-31, 2013 | Nov, 2013 (F) | Mar-Nov (in), 2013    |                     |
| Reedy Point, DE           | 34.3<br>(01/31/2014 20:54 UTC) | 59 F            | 52            | 38.6                  | 12 ft               |
| Annapolis, MD             | 30.7<br>(01/31/2014 20:54 UTC) | 60 F            | –             | 33.65                 | 14                  |
| La Grange                 | N/A                            | 58              | 57            |                       | 13.2 ft             |
| Talcony-Palmyra Bridgy NJ | 32.1<br>(01/31/2014 20:54 UTC) | 62              | 57            | 42.6                  | 12.3                |

#### 4.3.1 Preprocessing

Every cell content is preprocessed before it is sent to other components. We perform basic string cleaning such as the removal of :, 's, “, ‘ and non ASCII characters. We retain characters such as (, ), ^ to be used with NER and other components in the pipeline. For instance, *Acronym Tagger* uses () to identify the acronyms. We also split the node content using , and ;. The split content behaves as a new node content throughout the pipeline. For instance, a node content *New York, NY* will be split into two different strings *New York* and *NY*. However,

provenance information is retained that will be used while constructing the annotation graph (discussed in Section 6). We apply Stanford CoreNLP parser on the cell content to collect parser information such as POS tags. From this component, all noun phrases will be passed on to rest of the components.

### 4.3.2 Feature Taggers

*Feature Taggers* are an important set of taggers in the annotation pipeline. They handle several NLP and NER techniques to identify the features. The input to these taggers are the preprocessed tokens which are noun phrases (NP). In this section, we will discuss in detail each tagger, its corresponding dictionary (as applicable) and its role in the tabular data.

#### 4.3.2.1 Concept

A concept forms the leading entity or *talking* entity in the entire table or a subset of the table. In our example, there are three different concepts *Water Temperature*, *Precipitation* and *Water Level*. These concepts doesn't necessarily need to appear within a table. In certain cases, a concept will be available only in only one of the header cell. However, this concept may be applicable to the entire table. This situation often exists in tabular data such as annual report and other forms of statistical data. For example, in Table IV, the only concept present in the entire table is *Merchandise Shipments*. Other header cells such as *Total*, *Exports* and *Imports* come under (or describe further) about *Merchandise Shipments* but they do not directly add any meaning to their corresponding columns.

While the presence of these concepts are more common only in header cells, there are scenarios where the concepts are present in the *table caption* (See Section 5). Almost, all of

TABLE IV  
SINGLE CONCEPT FOR THE ENTIRE TABLE

| Merchandise Shipments | Total  | Exports | Imports |
|-----------------------|--------|---------|---------|
| Canada                | 67,810 | 21,095  | 46,715  |
| Mexico                | 85,604 | 25,348  | 60,256  |

these concepts are noun phrases whose constituents may be a noun singulars (NN) or noun plurals (NNS) along with some type of *modifiers* as described in the next section.

#### 4.3.2.2 Modifier

Modifier, as defined in English grammar<sup>1</sup>, is a word or set of words that add further meaning to a concept. In order to identify them from tabular data, we look for some important POS tags namely the adverbs (RB[RS]) and the adjectives (JJ[RS]). In our running example, the word *Recent* is a modifier to the concept *Water Temperature*. Under the pretext of unstructured data, the addition of adverbs to this list may seem irrelevant for describing a *concept* (a noun group). However, this has been found valid for the tabular data. For instance, consider a header cell labeled *Frequently flying jets*. The POS tags assigned for this caption is shown in Table V. We now notice that the label *Flying Jets* is a concept that contain the modifier *Frequently* which after stemming becomes *Frequent*.

---

<sup>1</sup><http://www.oxforddictionaries.com/us/words/grammar-a-z#modifier>

TABLE V  
ADVERBS AS A MODIFIER TO THE CONCEPT

|            |        |      |
|------------|--------|------|
| Frequently | flying | jets |
| RB         | VBG    | NNS  |

However, identifying these modifiers may not be accurate in several cases because of the nature of the text content and the inability of a standard NLP parser to perform well on tabular data. For instance, a cell content such as *Water Temperature in Great lakes* would have the word *Great* tagged as an adjective (JJ). This is completely misleading. To resolve such ambiguities, we give priority to entity recognition along with certain preprocessing in our annotation algorithm (Refer Section 4.4).

#### 4.3.2.3 Measurement Units and Symbols

This tagger is composed of a suite of dictionaries and pattern matching techniques. Standard units such as *mm*, *ft* and *celsius* are annotated by creating a custom dictionary that is based on the dictionary of units of measurements [22]. However, there are other symbolic representations that define a meaning to the data such as *>* for *greater than*, *<* for *less than* and so on. To deal with this, various matching mechanisms are used to identify units that go with values such as *Export (in billion dollars)*, *(in dollars)* and *12.4 (inches)*. The tagger will automatically convert the values based on the described units. For example, a data cell with value *1.2* whose

unit is identified as *million dollars* will convert the data cell to *1200000* and tag the unit as *dollars* to it. For every unit/symbol annotation, appropriate description will be retained from the dictionary. We compiled manually many such representations that are listed in Table VI. Further, these units or symbols does not necessarily appear in the *header cells*. However, because of the characteristics of a table, units that appear in a few of *data cells* in a column may be applied to all cells of that column. Initial dictionary is created using the dictionary of measurement units [22] as said earlier. The two main futuristic goals of annotating these units and symbols is (a) to enable the conversion between different units using the semantic graph that we will discuss in 6; (b) to semantically abstract the statistical data of a column.

TABLE VI  
SYMBOLIC REPRESENTATIONS AND THEIR DESCRIPTIONS

| Symbol template   | Description            | Presence            |
|-------------------|------------------------|---------------------|
| <                 | Less than              | Header or Data cell |
| >                 | Greater than           | Header or Data cell |
| +/-               | Approximately          | Data cell           |
| +/- {value}       | Error of               | Header cell         |
| (-)               | Has negative value     | Header cell         |
| %                 | Percentage             | Anywhere            |
| \$                | Dollars                | Anywhere            |
| in million        | multiply column values | Header cell         |
| in billion        | multiple column values | Header cell         |
| {value1}-{value2} | Range                  | Anywhere            |



#### 4.3.2.4 Uncertainty

Uncertainty or missing data is an issue in almost every table. Some *data cells* may be empty that should not be misrepresented with a value of  $\emptyset$ . It is important to recognize such cell content to abstract its accurate meaning automatically. Some commonly occurring content are *NA*,  $-$ ,  $\emptyset$ ,  $*$  and *Not available*. In certain special situations, it is also possible to predict the missing data. For example, if all of the cells in a data column has the a timestamp  $A$  except one (which is empty), then there is more probability that the same timestamp  $A$  can be used for the empty column. But, this assumption may not be valid for other types of data columns (such as *Average rainfall*) and we mark them as *NULL* by default.

#### 4.3.2.5 Spatial and Temporal

Any information available on the web are described using three important dimensions – Spatial, Temporal and Thematic [23]. In this section, we will discuss about the techniques involved in the identification and representation of spatial and temporal information present in the tables. We will also discuss about pattern matching and other NER techniques that are to identify and curate such information.

#### 4.3.3 Spatial

Tables may contain geographic information in a data column that may or may not have appropriate header in its *header cell*. For instance, in our running example, *Reedy Point, DE*, *Annapolis, MD* are locations. However, they don't have a proper header that describes them (1) as locations; (2) as location of type *Water station*. Thus, we see that there are two important issues to focus on – identification of spatial data in a table and recognizing the type

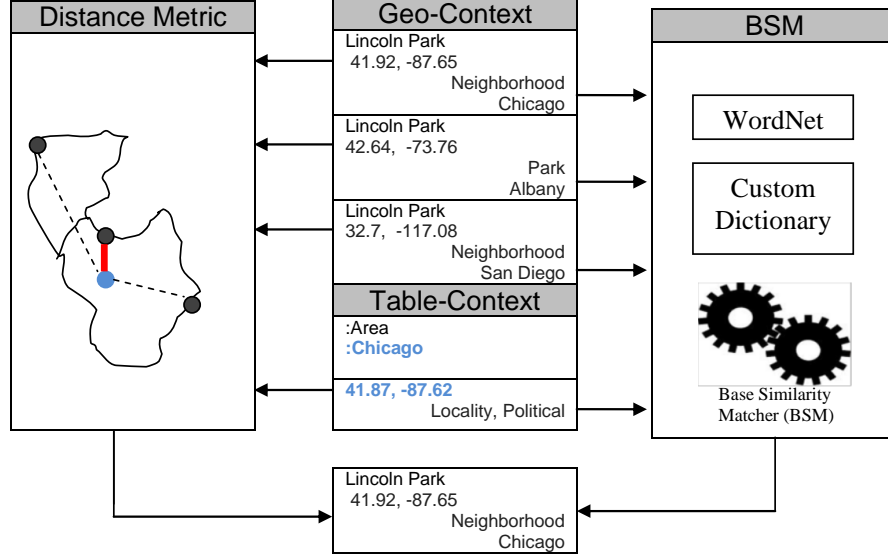


Figure 6. Disambiguation in Geocoding

of spatial data. The former can be further extended to the assignment of spatial coordinates such as *Latitude* and *Longitude*. This process of assigning the apposite geographic coordinates is referred to as *geocoding* [9]. Geocoding is also called as *GeoTagging* when dealing with media inputs such as images, videos or audios [24].

#### 4.3.3.1 Spatial Identification.

The challenges in spatial identification arises when multiple spatial context are present in a single cell as shown in our running example (the first column containing two types of spatial

information – *city* and *state*). In order to identify a spatial column, we first preprocess the cell content and perform named entity recognition on the *header cells* using a custom dictionary. This dictionary is created using the description of feature codes from GeoNames<sup>1</sup>. They also contain words such as territory, bay, aqueduct, lakes and battlefield. If the headers are as clear as these words, we finalize the type of spatial data in that column. Unfortunately, such clear headers are not so common. To resolve this, we use the complete GeoNames Gazetteer Data<sup>2</sup> to perform entity recognition on randomly sampled *data cells*. We do this entire process only on proper nouns (NNP[S]). While it is in fact true, as claimed earlier, that parser performs well only on a well phrased sentence, these less accurate POS tags still helps us to separate location data from the rest. For example, a flight status table may contain a *Status* column with words such as “Delayed” or “On time” in the data cells. This column can be easily eliminated as it would be annotated as a verb tense (VBP) and a preposition followed by a noun (PP NN), respectively.

#### 4.3.3.2 Geocoding.

While spatial identification helps to confirm that a particular text content is indeed a location (place name), geocoding helps to identify the geographic coordinates (latitude/longitude) of it after considering ambiguous place names. For instance, there are more than fifteen places with the name “Lincoln Park”. Some are parks, some are cities and some are small neigh-

---

<sup>1</sup><http://www.geonames.org/export/codes.html>

<sup>2</sup><http://download.geonames.org/export/dump/>

---

**Algorithm 1:** SpatialDisambiguationMatcher

---

```

begin
   $G \leftarrow$  List of GeoContext for each ambiguous place name ( $l_a$ )
   $t \leftarrow$  TableContext with unambiguous place name ( $l_u$ ), if available
   $D_1 \leftarrow$  Custom geo dictionary
   $D_2 \leftarrow$  WordNet
   $L \leftarrow \phi$  (disambiguated place name)
   $\Theta \leftarrow$  Similarity threshold
  Update  $G$  and  $t$  with definitions from  $D_1$  and  $D_2$  after removing stopwords
  for  $i = 1$  to  $sizeOf(G)$  do
     $\lfloor$  Compute  $sim(G_{l_i}, t)$ 
  if  $t$  contains  $l_u$  then
     $l_d \leftarrow$  closest location to  $l_u$ 
    if  $sim(G_{l_i=l_d}, t) > \Theta$  then
       $\lfloor$   $L \leftarrow l_d$ 
  if  $L == \phi$  then
     $\lfloor$   $L \leftarrow$  location in  $max(sim(G, t))$ 

```

---

borhoods within cities. Thus, in a table containing *list of parks in Chicago*, a data cell with the value “Lincoln Park” must be identified as the location of type “Park” and that park is in fact the one in Chicago. Geocoding is a wide research area that is still open for improvements [25, 26, 27, 28]. In this thesis, we propose a hybrid disambiguation approach using the Euclidean distance as well as the context information.

**Approach.** As Tobler’s first law states that “*Everything is related to everything else, but near things are more related than distant things*” [29], distance becomes an important metric in disambiguation. Further, in order to get additional evidence to disambiguate, context information is necessary. We produce two different kinds of context – GeoContext and TableContext. We use the complete GeoNames Gazetteer Data as discussed in the previous section. For each

location, we obtain the *latitude/longitude* coordinates along with the corresponding feature class (e.g., park, county, neighborhood or mountain), which forms the *Geo-Context*.

*TableContext* is a list of *concepts* so far identified. For better performance, geocoding is always performed as a last step. We build a custom matching algorithm called *SpatialDisambiguationMatcher* (*SpatialDM*) (See Algorithm 1) into AgreementMaker ontology matching system, which is extensible [30]. This algorithm combines the modified version of Base Similarity Matcher [31] and the Distance Metric. An example disambiguation process for “Lincoln Park” is illustrated in Figure 6. In a distance metric, the blue point indicates the unambiguous location and the thick red line indicates the shortest distance.

---

**Algorithm 2:** BuildAnnotationProfile

---

**Data:**  $\mathcal{T}_s$   
**Result:**  $\mathcal{T}'_s$   
**begin**  
  **forall** the data of  $\mathcal{T}_s$  **do**  
     $content \leftarrow Content(data)$   
    annotate  $content$  using annotation pipeline  
    update  $\mathcal{T}'_s$  with annotations  $(\mathcal{A}_c, \mathcal{A}_m, \mathcal{A}_t, \mathcal{A}_u, \mathcal{A}_v, \mathcal{A}_n)$   
  **for**  $L \in$  annotated lists in leafnodes **do**  
     $U_l \leftarrow RandomSample(L)$   
    annotate  $U_l$  using annotation pipeline  
    update  $U_l$  with annotations  $(\mathcal{A}_c, \mathcal{A}_m, \mathcal{A}_t, \mathcal{A}_u, \mathcal{A}_v, \mathcal{A}_n)$   
    **for**  $\mathcal{A}_x \in \{\mathcal{A}_u, \mathcal{A}_t, \mathcal{A}_l\}$  **do**  
      let  $l_i$  be an annotated element in set  $U_l$   
       $\mathcal{A}_x(L) = \arg \max_{\mathcal{A}_x} \Pr \{ l_1, l_2, \dots, l_n | \mathcal{A}_x \}$   
      update  $\mathcal{T}'_s$

---

#### 4.3.4 Temporal

Temporal tagger is responsible for the identification of temporal information represented in complex ways. We build a custom dictionary to identify and translate such text into a temporal data. Because of the nature of semi-structured data, we do not employ temporal normalization techniques [32] that are applied for unstructured text. For instance, it is rare to find relative temporal phrases such as *tomorrow* or *last week*. However, we use custom templates to identify ranges (e.g., Oct 15-26), time formats (e.g., UTC, GMT) and other commonly occurring phrases in tables. Besides the table, multiple temporal information may be present in other places such as table caption. In such cases, we retain both of them without resolving conflicts. This is further discussed in Chapter 5.

#### 4.4 Annotation Profile

In this section we will discuss an algorithm to build an annotation profile using the annotation pipeline. Algorithm 2 is performed on the tree  $\mathcal{T}_s$ . In the first phase, the entire tabular data is sent through the annotation pipeline to identify the appropriate annotations. Then, a set of randomly sampled element in the annotated *lists* (the data cells) of each leaf node in  $\mathcal{T}'_s$  is used to further update the nodes with the most probable annotation that may apply to entire list. For instance, in our running example, we see that the *units* are present only in some elements in the *lists* although that can be applied to all the elements. This second phase of the algorithm updates the origin leaf node with the annotation  $\mathcal{A}_u$  (a unit) along with the respective unit information. However, we limit this update only for  $\mathcal{A}_u$  and  $\mathcal{A}_l$  and  $\mathcal{A}_t$ .

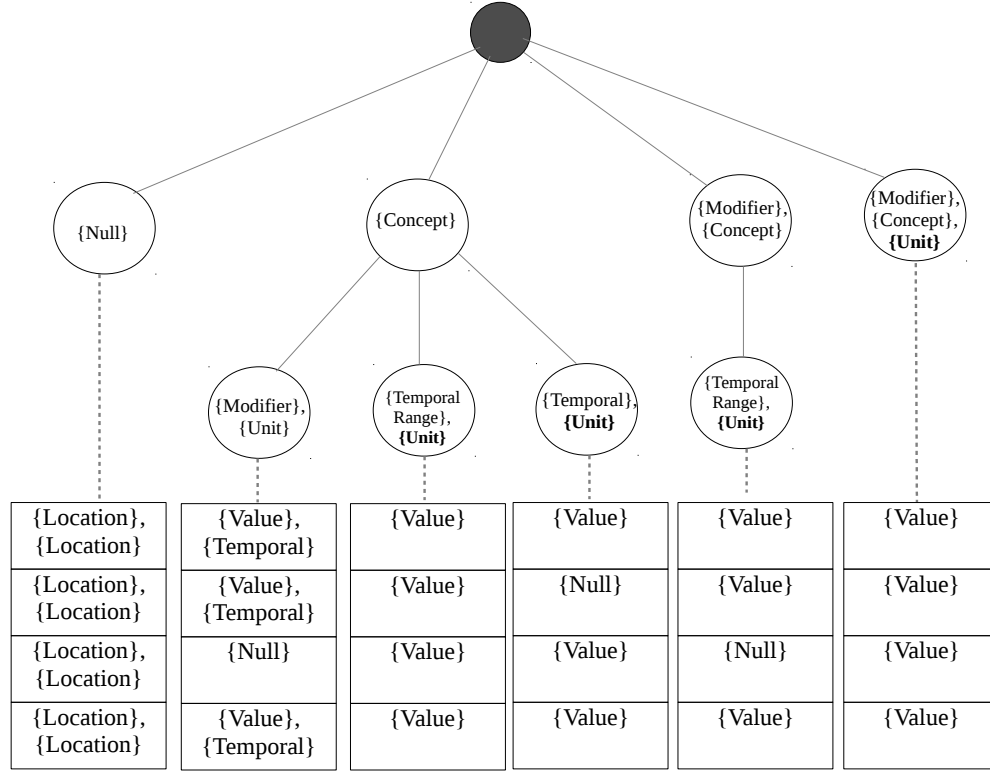


Figure 7. Annotation Profile

The complete annotation profile ( $\mathcal{T}'_s$ ) for our running example is shown in Figure 7. The annotations in *bold* are those identified using the *data cells*. We also find two *Location* annotations – state and city – in the first column. Further, each annotation will retain the original content of the cell as well as the annotation description. These will be used for the construction of the semantic graph (See Chapter 6).

## CHAPTER 5

### CAPTION ANNOTATION

Table captions contain critical metadata that describes a table in its entirety. Sometimes, the table may become meaningless without a caption. For instance, consider an example shown in Table VII that contain some important information related to *water levels* and *water stations*. But when the table alone is isolated from other metadata, we find that the terms *Station* and *Level* are completely obscure. Unless the system understands the caption *Table A: Water levels measured at water stations in Illinois*, the data within it is only comprehensible by humans.

TABLE VII

TABLE A: WATER LEVELS MEASURED AT WATER STATIONS IN ILLINOIS

| Station   | Level (ft) |
|-----------|------------|
| Brook, IL | 23.4       |
| Niles, IL | 19.8       |

#### 5.1 Characteristics

Table caption contain short phrases thereby making every word important. Unfortunately, they may not be a well phrased English sentences that can take a complete advantage of an NLP parser. However, we still use it mainly to identify its constituents and later resolve their



types in different ways. Another biggest challenge is present in the handling of table captions presented in title case (first letter of every word capitalized) similar to that of news headlines. This is a major obstacle to the NLP parser which does not allow proper identification of POS tags. The result of parsing such caption is shown in Table Table VIII. We see that the any word that starts with an upper case has been identified as Proper Noun (NNP) except for some unambiguous words such as *At* and *In*. Changing the complete string in to lower case will not work if it has a proper noun such as *Illinois*. Besides, the caption may also contain unwanted phrases such as *This table is about* or *Table A:*. To abstract semantics from these phrases, careful preprocessing is necessary before it can be sent to a parser.

TABLE VIII  
EFFECT OF TITLE CASE IN TABLE CAPTION

|       |        |          |    |       |          |    |          |
|-------|--------|----------|----|-------|----------|----|----------|
| Water | Levels | Measured | At | Water | Stations | In | Illinois |
| NNP   | NNP    | VBN      | IN | NNP   | NNPS     | IN | NNP      |
| Water | levels | measured | at | water | stations | in | Illinois |
| NNP   | NNS    | VBN      | IN | NN    | NNS      | IN | NNP      |

## 5.2 Preprocessing

In this section, we will look into some important preprocessing methods applied to table captions.

### 5.2.1 Title case

The preliminary focus was on finding a fix to the title case (placement of initial capital letters in all words of a title). We first perform an initial parse to classify the title of this type. After identifying a title case content, we use WordNet vocabulary (only nouns and verbs) to change the case except for the word that is present in the beginning and the word that appear after *semi-colons* (;). This process helps in correcting most of the common words. A similar fix was also presented for correcting the news headlines [33, 34].

### 5.2.2 Case refinement

Like the news headlines, most of the table captions contain one or more proper nouns (place names or organizations). However, some words may still be in an improper case thus taggers that use models (such as Stanford NER) do not perform accurately. To fix this, we perform gazetteer based NER to identify them at different levels:

**Location.** Locations are annotated using the spatial tagger used for the table annotations (Refer 4.3.2). Once a location has been identified, the caption is replaced with the case specific place name with proper case.

**Other proper nouns.** We use a dictionary of *rdf:labels* created using DBpedia [35] filtered by the *rdf:type* as *dbpedia-owl:Person* and *dbpedia-owl:Organisation*. This dictionary would also retain DBpedia resource URI for future use in entity linking (6). The annotated entities are replaced with their exact label.

The table caption after this stage free from ambiguous words. This will be used for the caption annotation (Section 5.3) and semantic graph construction (Chapter 6).

### 5.3 Caption Annotation

The caption is annotated using the pipeline discussed in 4.3. Along with those six types of annotations, we also retain POS tags as another layer of annotation to the entire caption. However, the identified annotation at this stage need not be accurate because of the nature of the content. For example, consider the caption discussed already in Table VIII. Our annotation pipeline detects “NN[S]” as concepts. Thus, from this caption, we will get *levels*, *water* and *stations* as captions whereas the correct concept hierarchy should link *water stations* and *levels*. We will see this process of meaningfully extracting and linking concepts in the next chapter.

## CHAPTER 6

### SEMANTIC GRAPH

Previous chapters covered various methods to identify complex metadata and data within the tables. However, in order to use them, the data needs to be semantically organized and modeled such that the resulting representation is able to provide a semantic description to them. In this chapter, we will introduce semantic graph and explain the methods to build it from the tabular data (Section 6.1). We will then discuss about an example graph created out of a table (Section 6.2) and about various other practical scenarios in the field of semantic web (Section Chapter 7.2.5).

#### 6.1 Semantic Graph

We describe a graph as a *semantic graph* if it has the ability to provide semantic information about any tiny piece of data. For example, let us consider a fully annotated table and caption for our running example on Table III as discussed earlier. If we take out one single *data cell* from it, say the cell containing *60 F*, then the graph should assist in creating a description such as “The *water temperature* at *Annapolis water station* in the state of *Maryland* for the time period *10-16-2013* to *10-31-2013* is *60 Fahrenheit*.”. This is possible by creating a simple ontology that has *hasA* relationships between every pair of nodes as shown in Figure 8. However, more complex ontology can be created depending on the purpose. Thus, we focus on creating a generic *triple* representation for each cell of a table as described below.

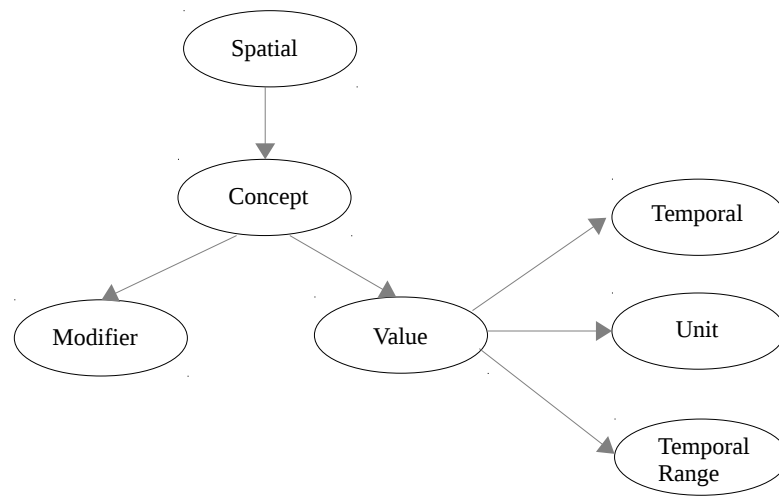


Figure 8. A simple ontology with *hasA* relationships

### 6.1.1 Triple Representation

In semantic web terminology, a *triple* is a subject-predicate-object expression that helps in the creation of statements. *Subject* denotes a resource (a “thing”), *predicate* may describe some property of the resource and create a relationship between a subject and an object, and *object* containing a value for the subject that may in turn be another resource. Thus, an object in one triple may become a subject for another thereby creating links between the resources. This can be realized from the illustration shown in Figure 9. These triples can be represented in Resource Description Framework (RDF), a widely used semantic web format. Other applications of these formats are discussed in Chapter 7.2.5.

```

dbpedia:owl:River rdfs:subClassOf dbpedia:owl:Stream
dbpedia:owl:Canal rdfs:subClassOf dbpedia:owl:Stream
dbpedia:owl:Stream rdfs:subClassOf dbpedia:owl:BodyOfWater

```

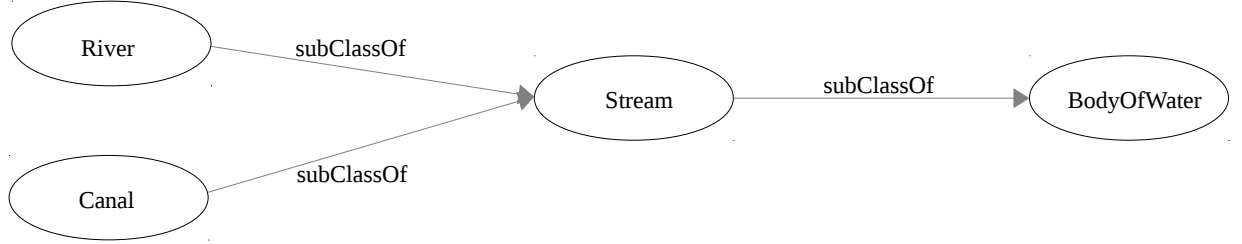


Figure 9. Triples as a graph

### 6.1.2 Construction

In this section, we will explain methods to construct a semantic graph from the annotated table graph profile  $\mathcal{T}'_s$  and from the annotated table caption.

#### 6.1.2.1 Table Graph

We describe two important steps in order to construct a semantic graph from the annotation profile of a table graph – Hierarchy Extraction and Linkage.

**Hierarchy Extraction.** The refined *header cells* of the graph  $\mathcal{T}'_s$  holds important metadata while retaining the original organization of the table. This allows us to extract the initial hierarchy. This hierarchy does not necessarily have to provide a meaningful ontology. For example, we cannot blindly create a *sub class* relationship between the parent and the child nodes of the graph. This can be realized by looking again at the annotation profile shown

TABLE IX

TABLE GRAPH - LINKAGE

| No.             | Rule  | Triple  |
|-----------------|---|---|
| 1               | $\text{Concept}_P \longrightarrow \text{Concept}_C$       | $\text{Concept}_C \text{ typeOf } \text{Concept}_P$                         |
| 2               | $\text{Concept} \longleftrightarrow \text{Unit}$          | $\text{Concept} \text{ hasUnit } \text{Unit}$                               |
| 3               | $\text{Concept} \longleftrightarrow \text{Temporal}$      | $\text{Concept} \text{ hasTemporal } \text{Temporal}$                       |
| 4               | $\text{Concept} \longleftrightarrow \text{TemporalRange}$ | $\text{Concept} \text{ hasTemporalRange } \text{TemporalRange}$             |
| 5               | $\text{Concept} \longleftrightarrow \text{Modifier}$      | $\text{Concept} \text{ hasModifier } \text{Modifier}$                       |
| 6               | $\text{Concept}_P \longrightarrow \text{Spatial}_C$       | $\text{Spatial}_C \text{ hasConcept } \text{Concept}_P$                     |
| 7               | $\text{Concept}_C \longrightarrow \text{Spatial}_P$       | $\text{Spatial}_P \text{ hasConcept } \text{Concept}_C$                     |
| 8 <sup>#</sup>  | $\text{Spatial}_P \longrightarrow \text{Spatial}_C$       | $\text{Spatial}_C \text{ hasConcept } \text{Concept}$                       |
| 9 <sup>#</sup>  | $\text{Temporal}_P \longrightarrow \text{Temporal}_C$     | $\text{Concept} \text{ hasTemporal } \text{Temporal}_C + \text{Temporal}_P$ |
| 10 <sup>#</sup> | $\text{Spatial} \longleftrightarrow \text{Concept}$       | $\text{Spatial} \text{ isA } \text{Concept}$                                |
| 11              | $\text{Value} \longleftrightarrow *$                      | $\text{Concept} \text{ hasValue } \text{Value}$                             |

<sup>#</sup> The concept will be extracted from the table caption (See 6.1.2.2)

in Figure 7 where we notice that the concept *Water Temperature* cannot make a *sub class* relationship with either the modifier *Recent* or the unit *Fahrenheit*. However, we can create a *HasA* relationship. We refer to this process of creating a proper semantic relationship as *Linkage* which is described below.

**Linkage.** After analyzing a considerable number of complex tables, we created a list of co-occurrence rules as shown in Table IX. The  $\longrightarrow$  indicates the *parent to child* connection between two nodes and the  $\longleftrightarrow$  represents simple co-occurrence which can occur even within a single node. The subscripts *P* and *C* indicates parent and child, respectively. Let us take a look at an example that satisfies rule 9 as shown in Figure 10. If we assume that a concept *Rainfall* has

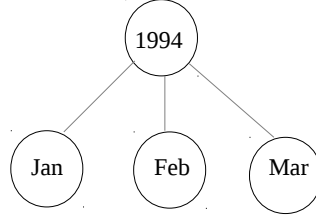


Figure 10. Structural Linkage

been identified from a table caption, this rule will create three triples as *Rainfall hasTemporal*  $\langle T1 \rangle$ , *Rainfall hasTemporal*  $\langle T2 \rangle$  and *Rainfall hasTemporal*  $\langle T3 \rangle$ . The  $\langle$  and  $\rangle$  indicates that the resources  $T1$ ,  $T2$  and  $T3$  contain the integrated information with the month and year. For instance,  $T1$  will contain data from both parent and child node – 1994 (year) and January (month) – respectively. Further, if there was a value in the *data cell*, then that value will be linked to the corresponding triple as per rule 11.

#### 6.1.2.2 Table Caption

In this section, we will look into the construction of the hierarchy from the table caption. Using the methods discussed in Chapter 5, we obtain an annotated table caption containing different semantic components. We will refine them further in three different steps. In the first step, we focus on extracting the concept ( $\mathcal{A}_C$ ) information from the caption. To do this, we parse the content with the Stanford parser to extract the parse tree. Then, we separate NPs that only contain the following POS tags: NN, NNS or NNP. These separate units will then become a concept. Let us look at an example parse tree for a caption as shown below:

```

(ROOT
  (S

```



```

(NP (NNP Water) (NNS levels))
(VP (VBD measured)
  (PP (IN at)
    (NP (NN water) (NNS stations))))
  (PP (IN in)
    (NP (NNP Illinois))))
(. .)))

```

From this tree, we obtain two noun phrases – *Water levels* and *water stations* – as potential concepts. The next two important steps are *Linkage* and *Hierarchy Extraction* similar to the one discussed in Section 6.1.2.1. Unlike the table graph where the initial structure is available, table caption doesn’t maintain any structure. Thus, we first perform *linkage* to semantically reduce the number of independent units and then apply methods to construct an ontology.

TABLE X

TABLE CAPTION - LINKAGE

| No | Rule  | Triple/Method  |
|----|---|--|
| 1  | $\mathcal{A}_M \Rightarrow \mathcal{A}_C$   | $\mathcal{A}_C$ hasModifier $\mathcal{A}_M$  |
| 2  | $\mathcal{A}_M \rightarrow \mathcal{A}_V \Rightarrow \mathcal{A}_C$                   | $\mathcal{A}_M$ hasValue CD/{*}<br>$\mathcal{A}_C$ hasModifier $\mathcal{A}_M$                     |
| 3  | $\mathcal{A}_{M_1} \rightarrow \mathcal{A}_{M_2} \rightarrow \mathcal{A}_C$           | $\mathcal{A}_C$ hasModifier $\mathcal{A}_{M_1}$<br>$\mathcal{A}_C$ hasModifier $\mathcal{A}_{M_2}$ |
| 4  | $\mathcal{A}_{T_1} \Rightarrow \text{TO}/\{\text{to}\} \rightarrow \mathcal{A}_{T_2}$ | Create TemporalRange<br>between $\mathcal{A}_{T_1}$ and $\mathcal{A}_{T_2}$                        |
| 5  | $\mathcal{A}_{M_1} \Rightarrow \mathcal{A}_{M_2}$                                     | Merge $\mathcal{A}_{M_1}$ and $\mathcal{A}_{M_2}$  |

$\rightarrow$  is followed by;  $\Rightarrow$  is immediately followed by; POS Tag/{word}

TABLE XI  
EXAMPLE - TABLE CAPTION LINKAGE

|                    |                 |                 |     |                 |     |                 |                 |
|--------------------|-----------------|-----------------|-----|-----------------|-----|-----------------|-----------------|
| <b>Caption</b>     | Daily           | river reports   | for | central         | and | lower           | north coasts    |
| <b>Annotations</b> | $\mathcal{A}_M$ | $\mathcal{A}_C$ | IN  | $\mathcal{A}_M$ | CC  | $\mathcal{A}_M$ | $\mathcal{A}_C$ |
| <b>Position</b>    | 1               | 2               | 3   | 4               | 5   | 6               | 7               |

**Linkage.** We perform linkage by defining cooccurrence rules between POS tags and the annotations. The list of rules is given in Table X. Let us look at an example caption *Daily river reports: central and lower north coasts*. The annotation we obtain is shown in Table XI. Using rule 1, we can combine the annotations at position 1 and 2. Using rule 3, we can combine the modifiers at position 4 and 6 with the concept at position 7.

TABLE XII  
TABLE CAPTION - HIERARCHY

| No | Rule   | Triple/Method   |
|----|--|---|
| 1  | $\mathcal{A}_{C_1} \Rightarrow \text{CC}/\ast \Rightarrow \mathcal{A}_{C_2}$                 | TABLE hasConcept $\mathcal{A}_{C_1}$ ; TABLE hasConcept $\mathcal{A}_{C_2}$ |
| 2  | $\mathcal{A}_{C_1} \rightarrow \text{IN}/\{\text{in,for,at}\} \rightarrow \mathcal{A}_{C_2}$ | $\mathcal{A}_{C_2}$ hasConcept $\mathcal{A}_{C_1}$                          |
| 3  | $\mathcal{A}_{C_1} \rightarrow \text{IN}/\{\text{with}\} \rightarrow \mathcal{A}_{C_2}$      | Create description for $\mathcal{A}_{C_1}$ with $\mathcal{A}_{C_2}$         |
| 4  | $\mathcal{A}_{C_1} \rightarrow \text{IN}/\text{of} \rightarrow \mathcal{A}_{C_2}$            | $\mathcal{A}_{C_1} \mapsto \mathcal{A}_{C_2}$                               |

$\rightarrow$  is followed by;  $\Rightarrow$  is immediately followed by;  $\mapsto$  simple directed edge; POS Tag/{word}

**Hierarchy Extraction.** We use a similar set of rules to extract the hierarchy from the caption. The list of rules is listed in Table XII. The rules are based on frequently occurring POS tag/word pair in table captions. Let us use the same table caption used for demonstrating *Linkage* here. Since hierarchy extraction uses the output of *Linkage*, we have two concepts namely “River reports” (linked with *Daily*) and “North coasts” (linked with *central* and *lower*). The resulting caption would now satisfy rule 2 which places “North coasts” ( $\mathcal{A}_{C_2}$ ) as a parent node linking to “River reports” ( $\mathcal{A}_{C_1}$ ).

The end result of this process is visualized in Figure 11. We can see the parent concept is *River Reports* which is known to describe the entire table. For uncertain or unknown *header cell*, this concept would be used as a *predicted* value. Further, the methods discussed above would be, sometimes, suitable for the content of a *data cell*. For instance, a typical census data would contain columns such as *Total families with own children* or *Number of people who are 16+ years old* and these cell values may be seen as a table caption.

## 6.2 Example

We will discuss the semantic graph (See Figure 12) created automatically for our running example shown in Table III. The subgraph containing thick ovals indicates that they are extracted from the table caption. The hierarchy is generated using the rules discussed previously. From this generic graph, we notice that each node is capable of providing a self description. The main reason to keep this graph generic is to allow flexible semantic modelling using semantic web formats such as RDF. For instance, one might be interested to create a new semantic

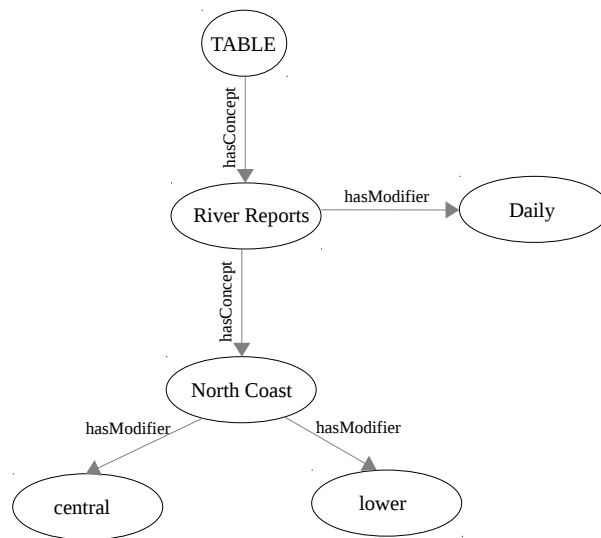


Figure 11. Table Caption - Semantic Graph

connection between *Water stations* and *Water level* by introducing a new class named *SubConcept* that may hold several useful axioms<sup>1</sup>. In fact, there are different methods to create such semantic connections automatically by using open data stores such as DBpedia<sup>2</sup> and Wikipedia [36]. Other potential uses are discussed in the next chapter.

---

<sup>1</sup>Axioms are formulas used in creating predicate logics that helps in reasoning. RDF/S vocabulary contains several axioms to help in this process. Refer <http://www.w3.org/TR/rdf-schema/>

<sup>2</sup><http://dbpedia.org/>

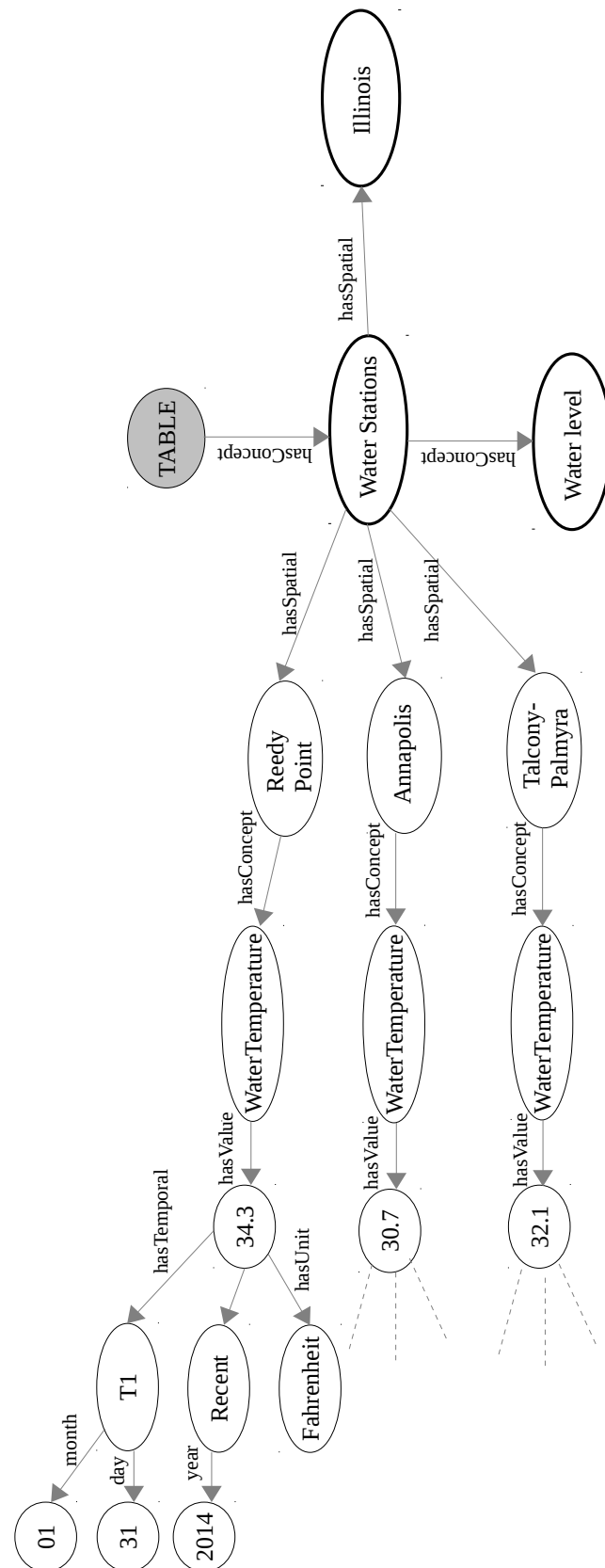


Figure 12. Complete Semantic Graph

## CHAPTER 7

### APPLICATIONS

The field of semantic web has shown an enormous growth in the recent years spanning over different areas such as biomedical [38], geospatial [37], information extraction [39] and even for publishing web content [40]. This is made possible mainly because of some of the powerful semantic web format such as RDF that contain rich vocabulary to semantically represent data from almost every domain. As discussed earlier, RDF data can easily be represented as triples. These triples can be stored in triple stores such as Virtuoso[41] or Sesame[42] and can be queried using the RDF query language called SPARQL[43]. In this chapter we will discuss about the potential application of our semantic graph on a semantic web format named *Data Cube* and on a semantic framework for Geospatial and temporal data Integration, Visualization and Analytics (GIVA).

#### 7.1 Data Cube

Several semantic web formats are being creating based on the RDF vocabulary to satisfy specific domain needs. For instance, Data Cube<sup>1</sup> is a semantic web format to represent a multi-dimensional data. Similarly, stRDF and stSPARQL have been created to represent geospatial

---

**Acknowledgement:** Part of the work in this chapter is presented elsewhere [37].

<sup>1</sup><http://www.w3.org/TR/vocab-data-cube/>

data [44]. In this section, we will discuss *Data Cube* vocabulary because of its high relevancy to the tabular data.

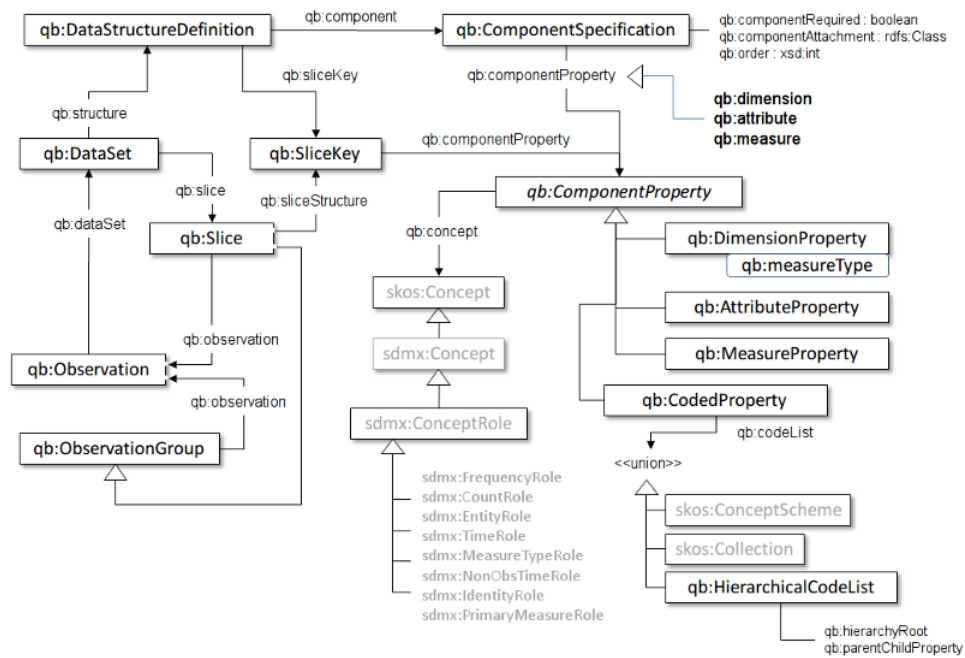
In the field of on-line analytical processing (OLAP), multi-dimensional data model is defined as “a variation of the relational model that uses multidimensional structures to organize data and express the relationships between data” [45]. Thus, we notice a close connection between the data in our semantic graph and multi-dimensional data. This can be realized by visualizing our running example (Table III) as a datacube having multiple spatial and temporal dimensions for a single concept. However, for the representation of semantic relationships and the meaningful hierarchy of our output graph, special vocabulary is necessary. Data Cube is one such RDF vocabulary that has been added into the W3C recommendation. It has rich vocabulary such that the statistical (multi-dimensional) data can be linked to other related data sets. Further, this model is compatible with the ISO standard cube model for sharing and exchanging statistical data called *Statistical Data and Metadata eXchange (SDMX)*<sup>1</sup> that is widely used in many organizations. The semantic graph can be used to create datasets using Data Cube in any manner as required. Because the Data Cube has an extensive vocabulary, we will only look into a few important features shown in Figure 13 to illustrate the usage of the graph.

**qb:Observation** A *class* to represent a single observation in the data cube.

---

<sup>1</sup><http://sdmx.org/>

<sup>1</sup><http://www.w3.org/TR/vocab-data-cube>

Figure 13. Data Cube Key Features<sup>1</sup>

**qb:measureType** A *property* this is a generic measure dimension and indicates the type of measure for a given observation (qb:Observation).

**qb:MeasureProperty** A *class* that holds the component properties which represent a value in an observation (qb:Observation).

**qb:concept** A *property* to hold the concept that is being measured.

One could clearly observe terms such as *qb:concept* and *qb:measureType* to be same as a *concept* and *unit* of the semantic graph, respectively. *qb:MeasureProperty* can be a *Concept*



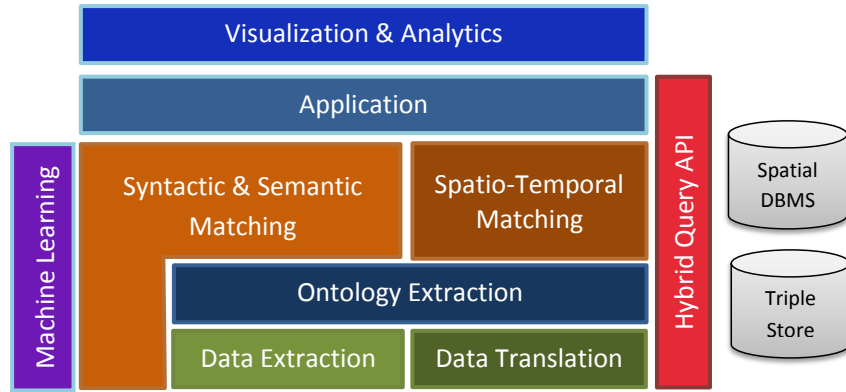


Figure 14. GIVA framework

combined with its *modifier*. Similarly, *qb:Observation* would be the *value* along with other information linked to it.

## 7.2 GIVA

GIVA is the acronym for Geospatial and temporal data Integration, Visualization, and Analytics. It is a semantic framework layered with different multifunctionality components. See Figure 14. The key feature of GIVA is its capability to deal with the the heterogeneity in geospatial data and their metadata. Some of the primary issues are (1) *heterogeneous file formats*, both standardized (e.g., Shapefile, KML, MapInfo TAB) and non-standardized (e.g., semi-structured data and flat files); (2) *lack of metadata* and *structural heterogeneity*, which stems from non-standardized file formats; (3) *heterogeneity in spatial and temporal resolutions*.

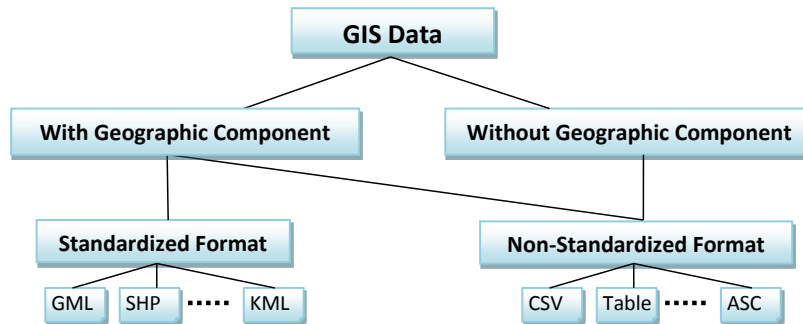


Figure 15. Hierarchy of spatial data types.

### 7.2.1 Data Extraction

Geospatial data is available on the web in a wide variety of formats, which can be systematically categorized as shown in Figure 15.

*Standardized* formats are those that are approved by OGC<sup>1</sup> and implement its standards. A *geographic component* present in these formats uses geodetic systems such as WGS84 that represent the geometric objects as, for example, polygons or polylines. *Non-standardized* formats include structured data (e.g., flat-files), semi-structured data (e.g., HTML tables, spreadsheets) and unstructured data (e.g., natural language content). The *geographic component* in these formats appear as raw text that requires the application of geocoding as discussed in 4.3.3 in order to uncover the geometric information. This *Data Extraction* component applies automatic methods tailored for these data formats and extracts the data in a unified format to be available for further processing by the components present in other layers.

---

<sup>1</sup><http://www.opengeospatial.org/standards/is>

### 7.2.2 Data Translation

The process of translating data from one file format to another is referred to as *Data Translation*. Proper abstraction of data formats is necessary for effective data integration [46]. In order to create geospatial mappings between these geospatial data, they need to be translated into a common spatial data format. One important issue with *non-standardized* formats, especially in semi-structured data, is to identify metadata such as column headers containing spatial coordinates or time stamps. The annotation methods discussed in Chapter 4 and Chapter 5 has the potential to assist in this process. For instance, it ensures that an unclear column header (e.g., coord) that contains geospatial coordinates (e.g., -82.16, 37.49) will be correctly identified as the column containing spatial coordinates and not mere numbers.

### 7.2.3 Ontology Extraction

The hierarchical characteristics of geospatial classification schemes can be modeled using a *part-of* or *is-a* relationship [31]. This component would also assist in extracting ontologies from relational tables, semi-structured data and structured data present in various sources. The work presented in this thesis plays a major role towards developing this component.

### 7.2.4 Matching

The semantic integration of geospatial data requires the identification of relations among concepts, properties, and data instances. This process, called *ontology matching*, uses syntactic and semantic characteristics of the ontologies to produce a list of *mappings*. Because of the organization of GIVA, this process also considers the spatial and temporal information while identifying a mapping. Two components present in this layer are described below:

#### 7.2.4.1 Semantic Matching

One of the prominent characteristics of ontologies is the presence of heterogeneity in their concept and structure. The mapping of such concepts among different ontologies requires specialized mechanisms that considers both syntactic and semantic information. We use AgreementMaker [30], a proven ontology matching system, that is capable of handling ontologies extracted from XML and RDF sources. Query expansion is performed by using the mappings produced by this tool [47]. Further, AgreementMaker uses machine learning techniques to assist in automatically adjusting the mapping configuration for better precision and recall [48].

#### 7.2.4.2 Spatio-Temporal Matching

Different geospatial data acquisition techniques introduce heterogeneity in spatial and temporal resolution. For example, data about the statistics of people affected by influenza may be recorded at a county level or at a city level. This problem is commonly referred to as *Modifiable Areal Unit Problem (MAUP)* [49]. GIVA deals with this problem by partitioning the spatial resolution into equal sized grids and by computing a weighted average. The size of a grid is automatically selected using machine learning techniques which also takes uncertainty into account. Heterogeneity in temporal resolution is also solved in a similar fashion using the *Hybrid Query API*. This spatio-temporal matching technique can be used for datasets about the same concept, for example *rainfall* or about different concepts, GIVA also allows the user to define a new dataset starting from datasets about different concepts. Suppose, for instance, that the user wants to build a dataset about *precipitation* starting from two datasets about *rainfall* and

*snowfall*. This can be achieved by adding the values of the two datasets and also by introducing an uncertainty value.

### 7.2.5 Storage Systems and Application

The framework has two different types of storage systems. In order to store and index the geographic information, a *Spatial DBMS* such as PostGIS<sup>1</sup> can be used. A *Triple Store* is used to store semantic information present in RDF or other semantic web formats. Virtuoso[50], OWLIM [51] are few examples of triple store. In order to facilitate a semantic geospatial query, a *Hybrid Query API* is used. The *Application* layer allows the installation of either web or stand-alone application to communicate with other framework components for enhanced user interaction. A sample web application based on GIVA framework is shown in Figure 16.

To summarize, GIVA enables the following functionality:

- The *Data Extraction* and *Data Translation* components handle the heterogeneity problems in geospatial file formats – standardized (e.g., GML, KML, SHP) and non-standardized (flat files, spreadsheets and HTML tables).
- The *Ontology Extraction* works along the two components below it in order to create a semantic web representation of data.
- Various spatial, temporal and metadata heterogeneities are handle by the two *matching* components.

---

<sup>1</sup><http://postgis.net/>

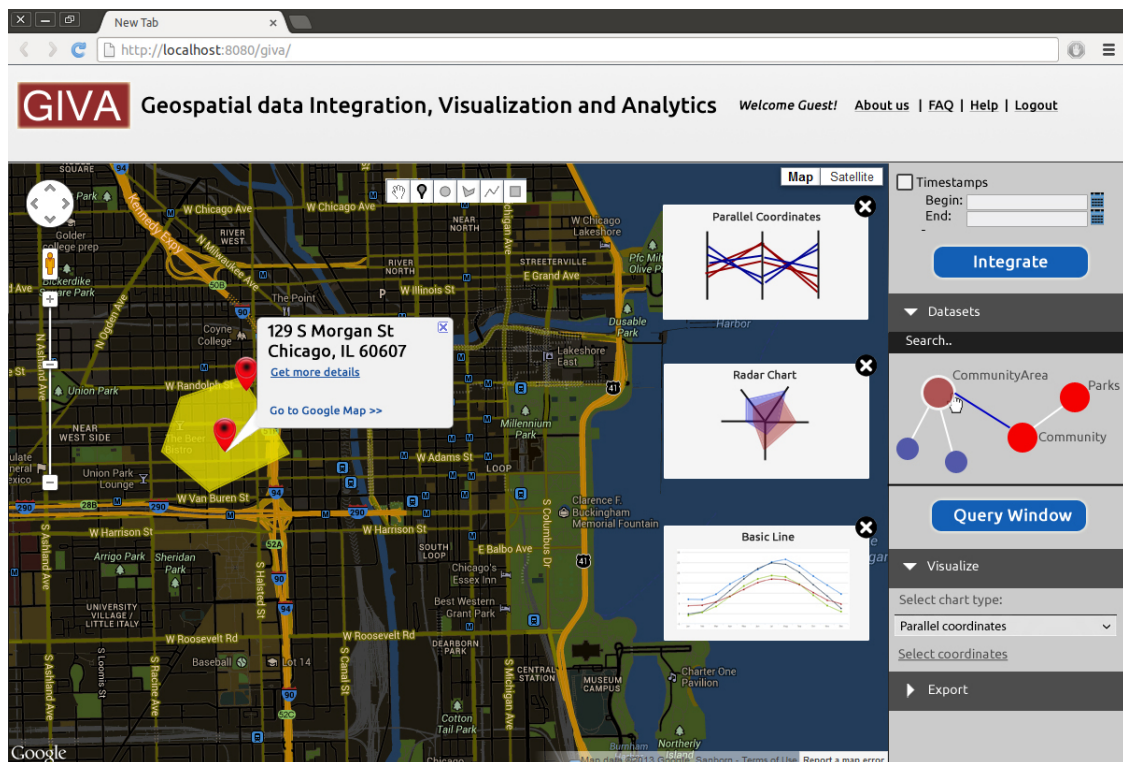


Figure 16. GIVA Application.

- The framework contains a hybrid data store and query API to coordinate between the semantic data and geospatial data.
- An *Application* assists in querying the processed data and to perform *Visualization and Analytics*.

## CHAPTER 8

### EXPERIMENTS

In this chapter, we discuss the datasets, experimental setup and evaluation of our approach on complex tables.

#### 8.1 Datasets

Since our focus is mainly on complex tables, we had to implement special techniques in order to retrieve them from the web. This is because the previous work were focussed on the extraction of tables that contain simple headers (See Chapter 9). Further, table search applications such as Google Tables (See 8.1.3.1) do not provide complex tables. In this section, we describe our approach to extract the complex tables from the web and also other test datasets for comprehensive evaluation.

##### 8.1.1 Complex Tables

A minimal list of web sites used for evaluation is listed in Appendix A. Among those sources, we picked randomly 250 tables in such a way that each table originate from different domain. Most of them contain statistical data or data that are published by government organizations. Our first goal was to filter out unwanted tables from the web pages such as those discussed in Section 2.1. To do this, we describe below a decision tree classifier incorporating different features that are necessary to identify *feature-rich* tables.

#### 8.1.1.1 Decision Tree

We use 10 different table features as listed in Table XIII. The training data for this decision tree model come from 100 heterogeneous sources that also include simple tables from Wikipedia<sup>1</sup> and Google Fusion contributing to a total of 1000 tables. Complex tables for training data include the spreadsheets that are converted to HTML. Since we focus on nested headers, we use the number of *colspan* as a feature instead of *rowspan*. However, during manual labeling of training data, the value for *Presence of rowspan* will always be *false* for a *valid* complex table. Type similarity is measured to identify the similarity among the data present in different columns. We do this by picking 10 rows through random sampling and dividing into two sets of rows *A* and *B*. For each set, we concatenate the content present in the columns. Thus, each set would contain a vector where index *i* contains the concatenated string corresponding to the column index *i*. The similarity index  $I_{sim}$  is computed as

$$I_{sim} = \frac{1}{n} \sum_{i=1}^n \frac{A_i \cdot B_i}{\|A_i\| \|B_i\|}, \text{ where } n \text{ is the number of columns in a table.}$$

We determine that the type similarity do exist (value is *true*) if the index is greater than 0.8 and does not exist (value is *false*) otherwise. All other features are straight forward analysis on the tables.

#### 8.1.2 Wikipedia tables

Wikipedia contains over a million relational tables that maintain clean structure, that is tables with headers in the first row and data in the rest. According to Venetis et al. [52],

---

<sup>1</sup><http://www.wikipedia.org/>



TABLE XIII  
DECISION TREE FEATURES FOR TABLE EXTRACTION.

| Feature  | Value                     |
|--|---------------------------|
| Number of Columns  | Integer                   |
| Number of Rows   | Integer                   |
| Number of <i>colspan</i>   | Integer                   |
| Presence of <code>&lt;th&gt;</code> in first two rows                                  | True, False               |
| Presence of <i>rowspan</i>   | True, False               |
| Existence of type similarity on randomly sampled data rows                             | True, False               |
| Background color difference between the rows   | True, False               |
| Font weight difference between the rows  | True, False               |
| Number of <code>&lt;img&gt;</code> or <code>&lt;object&gt;</code> tags in tables cells | S(1-3), M(4-10), L(>10)   |
| Average number of <i>characters</i> in <code>&lt;th&gt;</code> rows                    | S(5-10), M(11-20), L(>20) |

S, M, L = Small, Medium, Large

67% of information from these tables has a corresponding resource in YAGO [53] ontology thereby facilitating the process of linking open data more accurate. This is largely attributed to the quality of information present in the tables. Further, because of Wikipedia’s wider reception and participation in its crowdsourcing functionality, most of the information is up to date which is also well organized. While it was required to use a machine learning technique for extracting complex tables as discussed in 8.1.1, extracting tables from Wikipedia requires simple parsing as their HTML layout and their style information is uniform in every page containing table<sup>1</sup>. Thus tools such as *Google Tables* (Refer 8.1.3.1) would assist in fetching these tables. Nevertheless, our machine learning model will also detect these tables without feeding any such

---

<sup>1</sup>As of 2014, Wikipedia’s *table* tag uses the class *wikitable* and the schema row contains *th* tag.

layout information because of the way the model is trained. We extract 250 different Wikipedia tables by searching keywords such as *city*, *population*, *area etc..*

### 8.1.3 Captions

Table captions are extracted using a HTML tag priority based method. First, the feature-rich table location in a web page is identified using the machine learning method described in previous section. Then, a reverse parsing is performed from the marked location to identify the following prioritized HTML tags on first-come-first-serve basis:

1. <h1>
2. <h2>
3. <h3>
4. <b>

Since evaluation of table caption annotation and ontology modeling can be done independently from tables (except during the existence of uncertainty in table headers), we create a separate dataset for table caption using *Google Tables* for a comprehensive analysis on the performance of our annotation approach.

#### 8.1.3.1 Google Tables

Google Tables<sup>1</sup> is an experimental research product that facilitate table searching. Users enter keywords (similar to Google web search) and it delivers the list of web pages that contain

---

<sup>1</sup><https://research.google.com/tables>

related tables. However, these tables have simple and clear headers, that is, the tables have single row with *th* tags. Because of this limitation, we use these results only to extract table captions. We compile a list of keywords (See Table XIV) and extract 250 captions from the tables reported by this tool.

TABLE XIV  
KEYWORDS FOR TABLE SEARCH.

|                |             |
|----------------|-------------|
| climate        | export list |
| water          | properties  |
| temperature    | water flow  |
| population     | rivers      |
| sports         | animals     |
| teams          | hazard list |
| list of cities | plants      |

## 8.2 Evaluation

We evaluate our approach using the datasets described in previous section. For better clarity, we evaluate each component independently and discuss their results.

### 8.2.1 Table Annotation

Our table annotation approach using our annotation pipeline is evaluated on two datasets—Complex tables and Wikipedia tables. Each dataset (250 tables) contained approximately 1200 – 1800 features (data cells and header cells).

TABLE XV  
FEATURE DISTRIBUTION IN DATASETS.

| Dataset          | Concept | Modifier | Temporal | Unit | Value | Location | Total |
|------------------|---------|----------|----------|------|-------|----------|-------|
| Complex Tables   | 215     | 122      | 3324     | 266  | 655   | 145      | 1727  |
| Wikipedia Tables | 128     | 60       | 458      | 40   | 389   | 216      | 1291  |

**Gold standard.** We create gold standard by manually annotating features on both of the datasets. A detailed feature distribution is shown in Table XV.

**Baseline.** We use exact matching (ExMatch) as the baseline method for annotations. This method extracts the entire content within a cell (data or header) and uses the taggers from our annotation pipeline for entity recognition without using our algorithm. This method, however, does basic preprocessing such as removal of citations ([a]).

**Metric.** We use percentage accuracy of individual feature annotations to determine the quality of our annotation method. A correct annotation is defined as the annotation exactly defined by the gold standard. No partial score is given even in the case of combined annotations (e.g., Abraham Lincoln as two annotations – Abraham and Lincoln). This measure, thus, gives a complete picture of the annotation quality.

TABLE XVI  
PERCENTAGE ACCURACY OF TABLE ANNOTATION.

| Dataset             | Approach | Concept | Modifier | Temporal | Unit | Value | Location |
|---------------------|----------|---------|----------|----------|------|-------|----------|
| Complex<br>Tables   | AnnPipe  | 86.5    | 95.9     | 69.75    | 94.7 | 100   | 85.5     |
|                     | ExMatch  | 30.2    | 11.4     | 38.7     | 48.8 | 52.6  | 57.9     |
| Wikipedia<br>Tables | AnnPipe  | 97.6    | 93.3     | 97.3     | 85.0 | 98.9  | 96.3     |
|                     | ExMatch  | 76.5    | 90.0     | 89.5     | 45.0 | 93.8  | 91.6     |

The percentage accuracies of our annotation algorithm (Algorithm 2) and the baseline method are reported in Table XVI. We clearly notice that the baseline method performs poorly on complex tables but not on Wikipedia tables. The real challenge in *Complex tables* is the low quality content and poor organization. However, we notice our algorithm performing better in identifying most of the features. For example, *Modifier*, *Unit* and *Value* have an accuracy greater than 95%. The low performance in identification of *Location* is attributed to several

unwanted information present along with it. For instance, a table with only three *data cells* contained a non-standard abbreviation of cities such as *Alb*, *Chi* and *Temp* (referring to Albany, Chicago and Tempe). The same approach when applied on *Wikipedia tables*, performed well in identifying almost every features. This is mainly because of the quality of these tables as discussed in 8.1.2. The *Temporal* feature identification remained low in both approach because of its complex representations [54].

### 8.2.2 Caption Annotation

The test data containing 250 captions are used to evaluate the caption annotation and its ontology construction method. Table XVII reports the number of correct features identified along with its accuracy (recall). The drop in the accuracy is because of the natural language content. Although this evaluation used the same annotation pipeline used for *Table Annotation*, the absence of algorithm such as *BuildAnnotationProfile* reduces the accuracy of *Location*, *Temporal* and *Value* identification. Most of the errors in *Location* identification is attributed to our limited size of gazetteer data used for Named Entity Recognition. If the dictionary is refined further, the accuracy shall improve. *Unit* identification performed well in the captions. However, out of the 250 captions, there were only 15 units mentioned in the captions such as *Water temperature in celsius*. *Values* also remained less in captions. We often found the performance being affected by the presence of values and temporal information. For instance, dates being identified as values and vice-versa. We believe that refining the rules for identifying temporal information would improve accuracies of both temporal and value identification.

TABLE XVII  
PERCENTAGE ACCURACY OF CAPTION ANNOTATION.

| Annotation | Correct/Total | Accuracy (%) |
|------------|---------------|--------------|
| Concept    | 260/273       | 95.238       |
| Modifier   | 175/180       | 97.222       |
| Temporal   | 81/112        | 72.321       |
| Unit       | 14/15         | 93.333       |
| Value      | 70/80         | 87.5         |
| Location   | 123/180       | 68.333       |

### 8.2.3 Semantic Graph

The ontology construction is evaluated by manually looking at the *hierarchy* created by our methods. Rules discussed in Chapter 6 are taken as the base line. The application of these rules by the system is manually checked against the 250 captions. The system applied 378 rules out of which 312 rules were correct giving us an accuracy of 82.53%. Similarly, using the table structure and *Annotation profile*, we achieved 91%. We find this improvement is due to the presence of the table structure. The annotation graph and few links identified are shown in Figure 17.

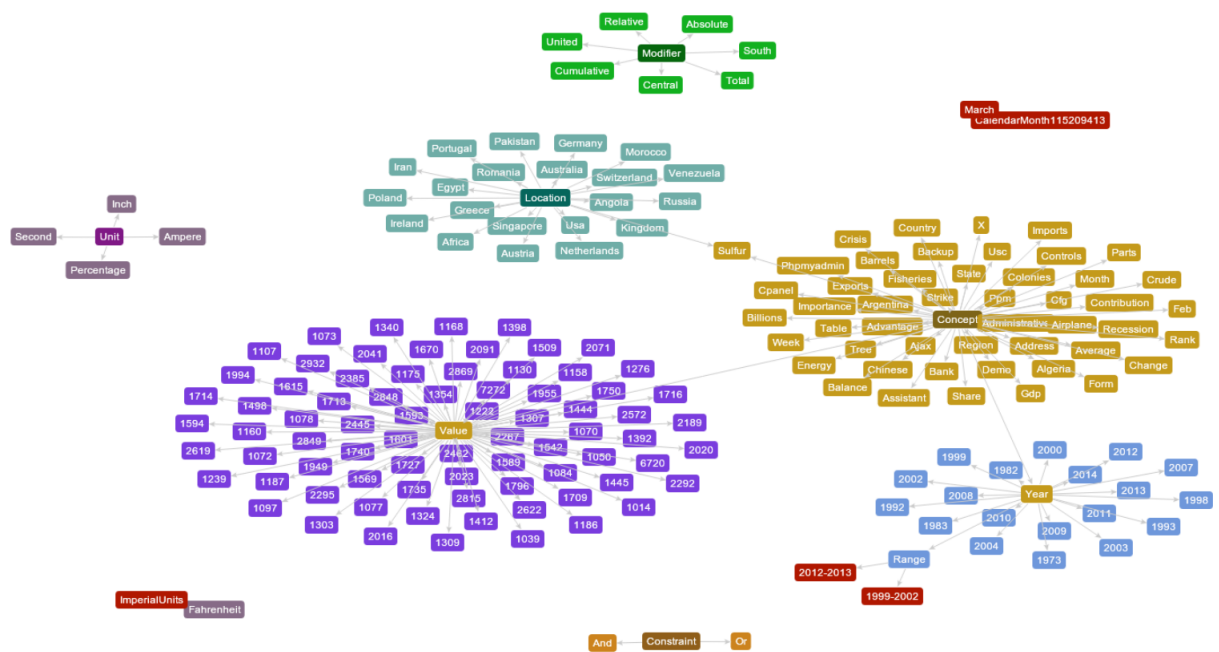


Figure 17. Annotation graph.



## CHAPTER 9

### RELATED WORK

Considerable work has been performed on the extraction of semantics from the tables [55, 56, 57, 58, 36, 59, 52] with a focus on generating semantic data. Venetis et al. [52] presented a method to recover the semantics from tables using *isA* relationship and proposed methods to assign appropriate labels for the table column headers. Knoblock et al. [56] propose a method to publish linked open data by looking at the semantics of a well structured table (KEGG pathway<sup>1</sup> data sources) and allowing users to modify the model to further refine its semantic description. Mulwad et al. [36] propose a framework to identify the type of a column header in such a way that the entire table (matrix) is involved in resolving a label and type of a single cell using a markov network. The semantic relevancy is computed using string matching algorithms such as Levenstein edit distance.

All of these research have been performed on simple tables and those that contain single *feature* per cell. These tables are equivalent to a table from a relational database. In such cases, there always exists a *isA* relationship between the data present in one column and its corresponding header. For instance, in a simple table shown in Figure 18, every cell has only one *feature* and the relations can be directly extracted between a *data cell* and a *header cell* using *isA* relations. Further, we also notice that the content of a *header cell* has a single feature (e.g., name or age as shown in Figure 18). This allows easy mapping of a *header cell*

---

<sup>1</sup>[www.genome.jp/kegg/pathway.html](http://www.genome.jp/kegg/pathway.html)

to a concept in a linked open data such as DBpedia [35] or YAGO [53]. For instance, *Virender Sehwag* maps to the DBpedia resource [http://dbpedia.org/page/Virender\\_Sehwag](http://dbpedia.org/page/Virender_Sehwag), *Batting Style* to the property <http://dbpedia.org/property/batting> and *Bowling Style* to the property <http://dbpedia.org/property/bowling>. Even if a property doesn't exist, defining labels is found relatively easy such as creating a property for *age*. Automatic tools such as Triplify [55] and D2R [57] have been developed to perform these operations. Unfortunately, not all tables, especially those that contain statistical data, are known to have simple and clean structures which has been discussed in previous chapters.

| Name                                 | Age | Batting style | Bowling style | Domestic team | Zone  |
|--------------------------------------|-----|---------------|---------------|---------------|-------|
| <a href="#">Mahendra Singh Dhoni</a> | 32  | Right-handed  | Right medium  | Jharkhand     | East  |
| <a href="#">Shikhar Dhawan</a>       | 28  | Left-handed   | Off break     | Delhi         | North |
| <a href="#">Gautam Gambhir</a>       | 32  | Left-handed   | Leg break     | Delhi         | North |
| <a href="#">Virender Sehwag</a>      | 35  | Right-handed  | Off break     | Delhi         | North |
| <a href="#">Murali Vijay</a>         | 29  | Right-handed  | Off break     | Tamil Nadu    | South |
| <a href="#">Cheteshwar Pujara</a>    | 26  | Right-handed  | Leg break     | Saurashtra    | West  |

Figure 18. Web Table with relational data.

## CHAPTER 10

### FUTURE WORK

Our approach towards representing highly complex tables creates a foundation for several future improvements and functionality.

**Semantic table search engine.** Major search engines such as *Google* or *Bing* fail to search within the tables as they are *keyword based*. While it has been reported in mid 2013 that Google has implemented a hybrid search (Google Hummingbird algorithm) by combining keyword and semantic techniques, they still work only on raw text data and not on tables.<sup>1</sup> Our semantic graph can be used in building up a semantic search engine on tables. Applications on sophisticated question-answering systems such as IBM Watson [38] can be developed using this semantic graph.

**Semantic web format.** While our work constructs a generic graph, it creates numerous possibilities to have it represented in a standard semantic web format that uses RDF/OWL. For instance, Data Cube (discussed in 7.1) is an appropriate RDF vocabulary to represent the generic graph with more properties and types. This enriches the semantics of table which can further assist in advanced semantic web uses such as reasoning [60]. Because of our modular annotation pipeline, additional features and taggers can be added to identify more types.

---

<sup>1</sup>[http://en.wikipedia.org/wiki/Google\\_Hummingbird](http://en.wikipedia.org/wiki/Google_Hummingbird)

**Annotation Quality.** The quality of annotation is another important area to focus on. This not only improves the annotation accuracy but also enhances the quality of semantic graph indirectly. To achieve this, more tables from different domains should be analyzed in order to add appropriate entries to the gazetteer for better functioning of NER built within the *Annotation Pipeline*. Further, more dictionaries can be easily added to the architecture to identify more features from the table. For instance, a new pattern can be added to identify a new temporal information such as *last week* or *this week*.

**Outside metadata.** Metadata present around the tables may sometimes assist in better understanding of the table. For example, a column header may have a *pinned* or *footnote* information, which can be present outside the table or at the end of the web page. While we did not come across tables like this more often, this functionality would definitely help in improving the table semantics.

**Linking Open Data.** The data represented in these semantic web formats can be linked to data present in a different dataset such as DBpedia [35]. This process linking the semantic data is commonly referred to as linking open data (LOD)<sup>1</sup>. While linking data from simple tables are well studied [36, 59, 52], there are several challenges in linking statistical data or data from complex tables to the concepts on the DBpedia or YAGO. For instance, a whole column containing numeric values with its corresponding column header as a year (such as 2009) will not have sufficient information to link to a concept from LOD database

---

<sup>1</sup><http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

using the approach described by Mulwad et al. [36] or Venetis et al. [52]. However, since our approach has the ability to connect that column to a concept within the table, linking can be done using our semantic graph.

Other related work that could further improve this research include the extraction of tables [6, 61, 62], table processing [63, 64] and identification of the schema (header rows or columns) from the tables [5].

## CHAPTER 11

### CONCLUSIONS

We introduced a novel approach to identifying different metadata present in semi-structured (tabular) data by resolving three important heterogeneity (structural, conceptual and metadata). First, we introduced our *Table annotation* and *Caption annotation* methodologies that can automatically identify different features present in a table. These methods use our *Annotation Pipeline* to accurately annotate the features using various NER and NLP techniques. In order to construct a semantic graph, we introduced an *Annotation Profile* and a list of ontology construction rules based on it. We evaluated our annotation approach and semantic graph construction rules on complex tables and Wikipedia extracted from the web using machine learning techniques. We achieved an average accuracy of 89% on annotation approach and an average accuracy of 86% on the construction of semantic graph. We also discussed the practical application of the constructed semantic graph on *Data Cube* vocabulary and its appropriate fit into our GIVA framework. We find that this approach towards processing semi-structured data to assist in semantic data integration and also as a foundation for creating a semantic tabular search engine.

## CITED LITERATURE

- [1] Cafarella, M. J., Halevy, A., Wang, D. Z., Wu, E., and Zhang, Y.: Webtables: Exploring the Power of Tables on the Web. PVLDB, 1(1):538–549, 2008.
- [2] Abiteboul, S.: Querying Semi-Structured Data. Springer, 1997.
- [3] Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H. F., and Secret, A.: The World-Wide Web. Communications of the ACM, 37(8):76–82, 1994.
- [4] Chaudhuri, S. and Dayal, U.: An Overview of Data Warehousing and OLAP Technology. ACM Sigmod record, 26(1):65–74, 1997.
- [5] Adelfio, M. D. and Samet, H.: Schema Extraction for Tabular Data on the Web. Proceedings of the VLDB Endowment, 6(6):421–432, 2013.
- [6] Gatterbauer, W., Bohunsky, P., Herzog, M., Krüpl, B., and Pollak, B.: Towards Domain-Independent Information Extraction from Web Tables. In Proceedings of the 16th international conference on World Wide Web, pages 71–80. ACM, 2007.
- [7] Chen, H.-H., Tsai, S.-C., and Tsai, J.-H.: Mining Tables from Large Scale HTML Texts. In Proceedings of the 18th conference on Computational linguistics-Volume 1, pages 166–172. Association for Computational Linguistics, 2000.
- [8] Cruz, I. F., Borisov, S., Marks, M. A., and Webb, T. R.: 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.
- [9] McCurley, K. S.: Geospatial Mapping and Navigation of the Web. In International World Wide Web Conference (WWW), pages 221–229. ACM, 2001.
- [10] Cruz, I. F., Ganesh, V. R., and Mirrezaei, S. I.: Semantic Extraction of Geographic Data from Web Tables for Big Data Integration. In Proceedings of the 7th Workshop on Geographic Information Retrieval, pages 19–26. ACM, 2013.
- [11] Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O.: Open Information Extraction for the Web. In IJCAI, volume 7, pages 2670–2676, 2007.
- [12] Fan, J., Kalyanpur, A., Gondek, D., and Ferrucci, D. A.: Automatic Knowledge Extraction from Documents. IBM Journal of Research and Development, 56(3.4):5–1, 2012.
- [13] Fader, A., Soderland, S., and Etzioni, O.: Identifying Relations for Open Information Extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1535–1545. Association for Computational Linguistics, 2011.
- [14] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A.: Unsupervised Named-Entity Extraction from the Web: An Experimental Study. Artificial Intelligence, 165(1):91–134, 2005.

- [15] De Marneffe, M.-C. and Manning, C. D.: The Stanford Typed Dependencies Representation. In Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation, pages 1–8. Association for Computational Linguistics, 2008.
- [16] Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B.: Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics, 19(2):313–330, 1993.
- [17] Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., and Soderland, S.: TextRunner: Open Information Extraction on the Web. In Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pages 25–26. Association for Computational Linguistics, 2007.
- [18] Joseph, T., Saipradeep, V. G., Ganesh, V. R., Srinivasan, R., Rao, A., Kotte, S., and Sivadasan, N.: TPX: Biomedical Literature Search Made Easy. Bioinformatics, 8(12):578, 2012.
- [19] Sundheim, B. M. and Chinchor, N. A.: Survey of the Message Understanding Conferences. In Proceedings of the workshop on Human Language Technology, pages 56–60. Association for Computational Linguistics, 1993.
- [20] Chieu, H. L. and Ng, H. T.: Named Entity Recognition: A Maximum Entropy Approach Using Global Information. In Proceedings of the 19th international conference on Computational linguistics-Volume 1, pages 1–7. Association for Computational Linguistics, 2002.
- [21] Settles, B.: Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets. In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, pages 104–107. Association for Computational Linguistics, 2004.
- [22] Rowlett, R.: How Many? A Dictionary of Units of Measurement, 2000. <http://www.unc.edu/~rowlett/units>.
- [23] Kim, K.-S., Zettsu, K., Kidawara, Y., and Kiyoki, Y.: 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.
- [24] Luo, J., Joshi, D., Yu, J., and Gallagher, A.: Geotagging in Multimedia and Computer Vision A Survey. Multimedia Tools and Applications, 51(1):187–211, 2011.
- [25] Overell, S. and Rüger, S.: Using Co-occurrence Models for Placename Disambiguation. International Journal of Geographical Information Science, 22(3):265–287, 2008.
- [26] Lieberman, M. D., Samet, H., Sankaranarayanan, J., and Sperling, J.: 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.



- [27] Leidner, J. L., Sinclair, G., and Webber, B.: 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.
- [28] Leidner, J. L.: Toponym Resolution in Text: “Which Sheffield is it?”. In International ACM SIGIR Conference (SIGIR), page 602. Citeseer, 2004.
- [29] Tobler, W. R.: A Computer Movie Simulating Urban Growth in the Detroit Region. Economic Geography, 46:234–240, 1970.
- [30] Cruz, I. F., Palandri Antonelli, F., and Stroe, C.: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. PVLDB, 2(2):1586–1589, 2009.
- [31] Cruz, I. F. and Sunna, W.: 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.
- [32] Martin, J. H. and Jurafsky, D.: Speech and Language Processing, 2000.
- [33] Chaumartin, F.-R.: UPAR7: A Knowledge-Based System for Headline Sentiment Tagging. In Proceedings of the 4th International Workshop on Semantic Evaluations, pages 422–425. Association for Computational Linguistics, 2007.
- [34] Loureiro, D., Marreiros, G., and Neves, J.: Sentiment Analysis of News Titles. In Progress in Artificial Intelligence, pages 1–14. Springer, 2011.
- [35] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In The semantic web, pages 722–735. Springer, 2007.
- [36] Mulwad, V., Finin, T., and Joshi, A.: 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.
- [37] Cruz, I. F., Ganesh, V. R., Caletti, C., and Reddy, P.: GIVA: A Semantic Framework for Geospatial and Temporal Data Integration, Visualization, and Analytics. In ACM Sigspatial International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS), 2013.
- [38] Ferrucci, D. A.: Introduction to “this is watson”. IBM Journal of Research and Development, 56(3.4):1–1, 2012.
- [39] Guha, R., McCool, R., and Miller, E.: Semantic Search. In Proceedings of the 12th international conference on World Wide Web, pages 700–709. ACM, 2003.
- [40] Buffa, M., Gandon, F., Ereteo, G., Sander, P., and Faron, C.: SweetWiki: A Semantic Wiki. Web Semantics: Science, Services and Agents on the World Wide Web, 6(1):84–97, 2008.
- [41] Erling, O. and Mikhailov, I.: Virtuoso: RDF Support in a Native RDBMS. In Semantic Web Information Management, pages 501–519. Springer, 2010.

- [42] Broekstra, J., Kampman, A., and Van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In The Semantic WebISWC 2002, pages 54–68. Springer, 2002.
- [43] Prud’hommeaux, E., Seaborne, A., et al.: SPARQL Query Language for RDF. W3C recommendation, 15, 2008.
- [44] Koubarakis, M. and Kyzirakos, K.: Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL. In The semantic web: research and applications, pages 425–439. Springer, 2010.
- [45] O’Brien, J. A. and Marakas, G. M.: Introduction to Information Systems, volume 13. McGraw-Hill/Irwin, 2005.
- [46] Abiteboul, S., Cluet, S., Milo, T., Mogilevsky, P., Siméon, J., and Zohar, S.: Tools for Data Translation and Integration. IEEE Data Engineering Bulletin, 22(1):3–8, 1999.
- [47] Cruz, I. F. and Xiao, H.: Ontology Driven Data Integration in Heterogeneous Networks. In Complex Systems in Knowledge-based Environments, eds. A. Tolk and L. Jain, pages 75–97. Springer, 2009.
- [48] Cruz, I. F., Fabiani, A., Caimi, F., Stroe, C., and Palmonari, M.: Automatic Configuration Selection Using Ontology Matching Task Profiling. In Extended Semantic Web Conference (ESWC), volume 7295 of LNCSE, pages 179–194, 2012.
- [49] Wong, W.: Principles of Two-Dimensional Design. New York, NY, Van Nostrand Reinhold Company, 1972.
- [50] Erling, O.: Virtuoso, a Hybrid RDBMS/Graph Column Store. IEEE Data Eng. Bull., 35(1):3–8, 2012.
- [51] Kiryakov, A., Ognyanov, D., and Manov, D.: OWLIM—A Pragmatic Semantic Repository for OWL. In Web Information Systems Engineering—WISE 2005 Workshops, pages 182–192. Springer, 2005.
- [52] Venetis, P., Halevy, A., Madhavan, J., Paşca, M., Shen, W., Wu, F., Miao, G., and Wu, C.: Recovering Semantics of Tables on the Web. PVLDB, 4(9):528–538, 2011.
- [53] Suchanek, F. M., Kasneci, G., and Weikum, G.: YAGO: A Core of Semantic Knowledge. In Proceedings of the 16th international conference on World Wide Web, pages 697–706. ACM, 2007.
- [54] Katz, G. and Arosio, F.: The Annotation of Temporal Information in Natural Language Sentences. In Proceedings of the workshop on Temporal and spatial information processing—Volume 13, page 15. Association for Computational Linguistics, 2001.
- [55] Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., and Aumüller, D.: Triplify: Lightweight Linked Data Publication from Relational Databases. In International World Wide Web Conference (WWW), pages 621–630. ACM, 2009.

- [56] Knoblock, C. A., Szekely, P., Ambite, J. L., Goel, A., Gupta, S., Lerman, K., Muslea, M., Taheriyani, M., and Mallick, P.: Semi-automatically Mapping Structured Sources into the Semantic Web. In International Semantic Web Conference (ISWC), pages 375–390. Springer, 2012.
- [57] Bizer, C. and Cyganiak, R.: D2R Server – Publishing Relational Databases on the Semantic Web. In International Semantic Web Conference (ISWC), page 26, 2006.
- [58] Dennis Jr, G., Sherman, B. T., Hosack, D. A., Yang, J., Gao, W., Lane, H. C., and Lempicki: DAVID: Database for Annotation, Visualization, and Integrated Discovery. Genome Biology, 4(5):P3, 2003.
- [59] Syed, Z., Finin, T., Mulwad, V., and Joshi, A.: Exploiting a Web of Semantic Data for Interpreting Tables. In Web Science Conference, 2010.
- [60] Wang, X. H., Zhang, D. Q., Gu, T., and Pung, H. K.: Ontology Based Context Modeling and Reasoning Using OWL. In Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, pages 18–22. IEEE, 2004.
- [61] Jannach, D., Shchekotykhin, K., and Friedrich, G.: Automated Ontology Instantiation from Tabular Web SourcesThe AllRight System. Web semantics: science, services and agents on the world wide web, 7(3):136–153, 2009.
- [62] Liu, Y., Bai, K., Mitra, P., and Giles, C. L.: TableSeer: Automatic Table Metadata Extraction and Searching in Digital Libraries. In Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, pages 91–100. ACM, 2007.
- [63] Limaye, G., Sarawagi, S., and Chakrabarti, S.: Annotating and Searching Web Tables Using Entities, Types and Relationships. Proceedings of the VLDB Endowment, 3(1-2):1338–1347, 2010.
- [64] Embley, D. W., Hurst, M., Lopresti, D., and Nagy, G.: Table-Processing Paradigms: A Research Survey. International Journal of Document Analysis and Recognition (IJ DAR), 8(2-3):66–86, 2006.

## APPENDICES

## Appendix A

### TABLE SOURCES

1. <http://www.bls.gov/news.release/ximpim.nr0.htm>
2. [http://codex.wordpress.org/Backing\\_Up\\_Your\\_Database](http://codex.wordpress.org/Backing_Up_Your_Database)
3. <http://data.worldbank.org/indicator/IC.EXP.DOCS>
4. [http://en.wikipedia.org/wiki/Foreign\\_trade\\_of\\_Argentina](http://en.wikipedia.org/wiki/Foreign_trade_of_Argentina)
5. <http://export.farnell.com/>
6. <http://export.gov/>
7. <http://www.adobe.com/legal/compliance/export.html>
8. [http://www.bls.gov/news.release/archives/ximpim\\_02142014.htm](http://www.bls.gov/news.release/archives/ximpim_02142014.htm)
9. <http://www.daff.gov.au/export/meat/meat-commodity-export-permit-requireme>
10. [http://www.deadiversion.usdoj.gov/imp\\_exp/doc/](http://www.deadiversion.usdoj.gov/imp_exp/doc/)
11. <http://www.eia.gov/country/cab.cfm?fips=IR>
12. <http://www.geonames.org/export/codes.html>
13. <http://www.infoplease.com/ipa/A0104811.html>
14. <http://www.law.cornell.edu/uscode/text/22/chapter-39>
15. <http://www.metal-pages.com/resources/chinese-export-tariffs/>
16. [http://www.nirsoft.net/utils/dll\\_export\\_viewer.html](http://www.nirsoft.net/utils/dll_export_viewer.html)

## Appendix A (Continued)

17. <http://www.sba.gov/about-offices-content/1/2889/resources/14315>
18. <http://www.sba.gov/content/state-trade-and-export-promotion-step-fact-sheet>
19. <http://www.theguardian.com/news/datablog/2010/feb/24/uk-trade-exports-imports>
20. <http://www.unzco.com/basicguide/c11.html>
21. <https://groups.drupal.org/node/21338>
22. <http://apps.fas.usda.gov/export-sales/esrd1.html>
23. <http://www.nrc.gov/reading-rm/doc-collections/cfr/part110/full-text.html>
24. <https://www.federalregister.gov//revisions-to-the-export-administration-regulations>
25. <http://www.eia.gov/country/cab.cfm?fips=IR>
26. <https://www.ecb.europa.eu/mopo/eaec/trade/html/index.en.html>
27. <http://data.worldbank.org/indicator/LP.EXP.DURS.MD>
28. <https://www.library.ca.gov/CRB/97/10/crb97010.html>
29. <https://www.federalregister.gov//revisions-to-the-export-administration-regulations>
30. <http://endnote.com/en/online-databases>
31. [http://www.eia.gov/dnav/pet/pet\\_move\\_exp\\_dc\\_nus-z00\\_mbbldpd\\_a.htm](http://www.eia.gov/dnav/pet/pet_move_exp_dc_nus-z00_mbbldpd_a.htm)
32. <http://www.vandyke.com/download/export.html>
33. <http://www.for.gov.bc.ca/het/export.htm>

## Appendix B

### TABLE CAPTION

- Climate of 100 Selected U.S. Cities
- Water Levels
- U.S. ARMY CORPS OF ENGINEERS, ST. LOUIS DISTRICT, RIVER & RESERVOIR  
DAILY REPORT
- Monitored Water Supply Reservoirs
- Index of Sites - Tides, Currents, and Water Levels
- All State Park Current Conditions
- List of Contaminants and their (MCLs)
- Daily Reservoir Storage Summary
- Table 1. Physical properties for solid cylinder unknowns.
- Lake dimensions
- Freshwater sources (top 15 countries)
- Perth dam locations
- White River Reservoir
- Short term and periodic changes
- Water levels

## Appendix B (Continued)

- Aquatic Life Criteria Table
- Table with minimum depths on the Maritime Danube
- Daily river reports: Central and Lower North Coasts
- Chronological Listing of Hudson River School Painters
- RB1: Embankment - Woolwich Arsenal
- RB2: Bankside - Embankment - Millbank - St George Wharf (Tate to Tate and St George Wharf)
- RB3: London Bridge - Canary Wharf (for fares see route RB1)
- Oregon River Flows



## VITA

Venkat Raghavan Ganesh Sekar was born in Vellore, India. He completed his bachelor's degree in Computer Science and Engineering at *SASTRA University*, India, in 2009 following which he worked as a *corporate mentor* at Tata Consultancy Services, Trivandrum, India for a brief period. He then joined TCS Innovation Labs at Hyderabad, India where he made his contributions to conference papers and an international patent. Later, he joined *University of Illinois at Chicago* for his master's degree in Computer Science starting Fall 2012, where he has been a Research Assistant to Dr. Isabel Cruz, Professor of Computer Science.

### Publications/Patents

1. Isabel F. Cruz and Venkat R. Ganesh, *Semantic Extraction of Geographic Data from Web Tables for Big Data Integration*, In ACM GIS SIGSPATIAL Workshop on Geographic Information Retrieval (GIR), 2013
2. Isabel F. Cruz, Venkat R. Ganesh, Claudio Caletti and Pavan 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), pages 554-557, 2013
3. Thomas, J., Saipradeep, V., and Venkat R. Ganesh, *Automated Dictionary Creation for Scientific Terms.*, Patent No. 13/752,60 (U.S.) 2013
4. Thomas, J., Saipradeep, V., and Venkat R. Ganesh, Rajgopal, S., and Aditya, R., *TPX: Biomedical Literature Search Made Easy.*, Bioinformation 8, no. 12 (2012): 578