Chapter

TULIP: A Five-Star Table and List - From Machine-Readable to Machine-Understandable Systems

Julthep Nandakwang and Prabhas Chongstitvatana

Abstract

Currently, Linked Data is increasing at a rapid rate as the growth of the Web. Aside from new information that has been created exclusively as Semantic Web-ready, part of them comes from the transformation of existing structural data to be in the form of five-star open data. However, there are still many legacy data in structured and semi-structured form, for example, tables and lists, which are the principal format for human-readable, waiting for transformation. In this chapter, we discuss attempts in the research area to transform table and list data to make them machine-readable in various formats. Furthermore, our research proposes a novel method for transforming tables and lists into RDF format while maintaining their essential configurations thoroughly. And, it is possible to recreate their original form back informatively. We introduce a system named TULIP which embodied this conversion method as a tool for the future development of the Semantic Web. Our method is more flexible compared to other works. The TULIP data model contains complete information of the source; hence it can be projected into different views. This tool can be used to create a tremendous amount of data for the machine to be used at a broader scale.

Keywords: data labeling, knowledge discovery, knowledge representation, Linked Data, open data, semantic annotation, Semantic Web

1. Introduction

The web has evolved through many stages. Contents on the Web have been changed in both form and method. Searching for information on the Web with keywords is very limited. In the future, we will have a lot of information, causing the traditional search to be insufficient. To perform a better search, the semantic of information must be exploited. One way to represent such concept is to use Linked Data. The conversion of already abundant data into the Linked Open Data format will allow the intelligent search. This extension technology of the Web is called the Semantic Web.

This chapter will briefly introduce the Semantic Web and underlying technologies, including the fundamental element of the Semantic Web called Resource Description Framework (RDF). Currently, there are many different forms of information on the web. So, there is a lot of research related to the conversion of these

contents into a searchable format by the Semantic Web. This chapter discusses open data standards and the making of machine-understandable data.

Many forms of conversion have already been proposed by many research (we will review them in Section 3). However, the conversion of tables and lists is still problematic. We propose a novel method to convert tables and lists to five-star open data with the data model called TULIP. The following sections discuss TULIP vocabulary as well as brief examples of its application.

2. Background

Before mentioning the Semantic Web, it is useful to describe the development of the Web in a nutshell. Nova Spivack has explained Web 3.0, the latest development of the Web [1]. The first era is Web 1.0 which consist of contents that can rarely be changed, most of which are generated by research institutions and business organizations. The next era, Web 2.0, has contents that can be changed frequently, and most of them come from the users creating and updating their information, such as Weblog (blog), wiki, social networks, etc. Now we are in the era of Web 3.0 and Semantic Web. It focuses on linking the data between computers together and processing them directly by computers.

2.1 Structured, semi-structured, and unstructured data

In terms of simplicity of data processing by computers, the data can be classified into three types:

1. Structured data is the data that has a definite structure, such as data contained in relational databases. This type of data can be directly processed.

2. Semi-structured data is the data that cannot be wholly identified for its structure, such as a table, list, chart, etc. Although humans can see these data as "structured" and can easily understand them, it is not possible for computers to manipulate these data directly because of uncertainty and ambiguity in terms of structure and meaning. It is necessary to convert them by means of various methods before further processing.

3. Unstructured data is the data that has no simple structure, such as text in the form of essays, pictures, audio, video, etc. They must be preprocessed by specific methods, such as natural language processing (NLP) and other methods to convert them into a format that can be manipulated by computers. This type of data has the highest uncertainty and ambiguity.

2.2 What is the semantic web

Indeed, the Semantic Web is not all new technology. Tim Berners-Lee, who invented the World Wide Web in 1990 [2], announced the concept of Semantic Web in 2001 in the *Scientific American* article [3]. Semantic Web is an extension of the Web that we currently use in which information is given well-defined meaning. In other words, Semantic Web is a Web of data that can be processed directly or indirectly and "understood" by computers. Steve Bratt, CEO of World Wide Web Consortium (W3C) [4], contrasts the World Wide Web which uses hyperlinks to link various resources between computers connected by the Internet and Semantic Web which uses relationship or "meaning" to link resources or "objects" together.

Each object in the Semantic Web is a part of a huge distributed database on the Internet which can be processed by computers, and results can be presented in a variety of formats as required by users.

In summary, Semantic Web is a technology that is based on the current technology and the Internet. It relies on a set of protocols at different levels that works together to create the distributed data structure on the Web in the form of relationships that linked together across the system through the Internet. An example of the benefits of the Semantic Web is to search for information about proteins that affect the treatment of Alzheimer's disease as currently being studied around the world. If searching using a regular search engine, it may reach about 223,000 documents around the Web. Many of these documents may not be relevant. However, if searching through the Semantic Web, the result is the list of 32 proteins from the Semantic Web of researchers sharing and exchanging information on the disease.[1]

2.3 Elements of the semantic web

As with other services on the Internet, most of which is the integration of standard or commonly used components. In the case of Semantic Web, it consists of various components such as Unicode, Uniform Resource Identifier (URI), Extensible Markup Language (XML), and other standards. Some frameworks have been developed, improved, or modified from the existing ones, such as the Resource Description Framework (RDF), RDF Schema (RDFS), Web Ontology Language (OWL), and SPARQL Protocol and RDF Query Language (SPARQL). In this chapter, we mainly focus on RDF and SPARQL.

Resource Description Framework (RDF) is the main structure for storing the smallest components of facts in the knowledge base linked within the Semantic Web. Basically, an RDF is a "sentence" that has three parts: a subject, a verb (or predicate), and an object. Both subject and object will be the identity, i.e., the name of the resource in the form of a URI (in the case of the latter, it can be literal or constant). A predicate (also in the form of URI) describes the relationship between them. These sentences are called RDF triples. The triples are linked together as a graph structure called the RDF graph, which is sometimes referred to as the semantic graph or knowledge graph.

2.4 What is linked data and five-star open data

It is said that Semantic Web, though simple, is still not being used extensively [5]. Linked Data is a set of guidelines for disclosing, sharing, and connecting pieces of information or knowledge on the Semantic Web using URI and RDF [6]. The Linked Open Data (LOD) project by Chris Bizer and Richard Cyganiak aims to expand the web with shared data by distributing open datasets in the RDF format on Semantic Web and creating the RDF links between these datasets [7]. A class of open data sharing level is defined as the number of stars (★) as follows:[2]

★One-star level has the only requirement to make the information public in any data format.

★★Two-star level has a provision that the disclosed information must be in a format that is not unstructured data, whether it is a proprietary format or not.

---

[1] https://www.ted.com/talks/tim_berners_lee_the_next_web/transcript

[2] https://5stardata.info/en/

★★★Three-star level requires that the data must be in a structured form with an open standard format.
★★★★Four-star level determines that the data must be in an open standard in the Semantic Web format, such as RDF.
★★★★★Five-star level requires that the data must be linked to other open data to be a complete Linking Open Data.

3. Towards machine-understandable data

There are many works related to transforming data from tables and lists to Linked Data. Some research involves extracting table-type data in various formats such as spreadsheets, relational databases, etc. and then converting them to RDF data. Those researches can be divided into groups as follows.

3.1 Research related to the creation of facts into linked data

There are many research works related to filling facts into Linked Data. The most discussed projects [8–11] are DBpedia, YAGO, Freebase, Wikidata, and OpenCyc. There are also several related researches which can be divided into groups as follows:

3.1.1 Extracting facts from various parts of Wikipedia

DBpedia is a joint research of the Free University of Berlin and Leipzig University in Germany [12]. The objective is to extract Wikipedia structured data such as infoboxes and categories including some unstructured data such as abstracts [13]. DBpedia supports Wikipedia information in many languages [14]. The goal of this project is to be the core to link other datasets of Linked Data together [15]. The result is a core that has more than 3 billion facts about 4.58 million topics which are divided into nearly 600 million facts from English Wikipedia articles, and the remainder is more than 2.5 billion facts from Wikipedia in other languages.

With the limitation of extracting data from tables, such as how to classify different types of tables and assigning names to them, DBpedia then chooses to extract only the structured data [13]. However, there are additional capabilities in the later version of the framework for extracting table data in Wikipedia, but it only extracts the table data as HTML tag block, not as the RDF triples that can be directly queried using SPARQL [12]. DBpedia has encouraged the development of algorithms to extract data from tables and lists in Wikipedia using the DBpedia framework by proposing a project in the Google Summer of Code (GSoC) from mid-2016 which continued to develop the project until 2017.[3] It has yet to publish the relevant academic work and has not yet been implemented in the latest version of DBpedia framework.

Isbell and Butler published a research paper created at HP's Digital Media Systems Laboratory [16]. They conducted a study of the conversion of data from Wikipedia structured infoboxes and has some parts that cover semi-structured and unstructured data.

YAGO is a research project from the Max Planck Institute for Informatics in Germany [17]. The objective is to extract structured data from Wikipedia categories by applying Synsets data from Princeton University's WordNet project. The project contains 120 million facts in 10 million topics.

[3] https://wiki.dbpedia.org/blog/dbpedia-google-summer-code-2016 DBpedia @ GSoC 2016

BabelNet [18] and Multilingual Entity Taxonomy (MENTA) [19] extract facts from Wikipedia and WordNet as well as YAGO, but BabelNet and MENTA aimed at creating a multilingual knowledge base.

3.1.2 Manually recording facts into the knowledge base

Freebase of Metaweb Technologies [20] is a Web-based knowledge base where users share structured information directly through a Webpage specifically designed for recording and verifying information [21] (unlike DBpedia and YAGO, in which structured data was converted from Wikipedia.) After being acquired by Google in 2010, its data was transferred to Wikidata in 2014. Finally, in 2016, Freebase was closed, and it has been integrated into the Google Knowledge Graph. It is later being developed into Knowledge Vault: a Google research that aims to create an automated process to build the knowledge base directly from the Web [22].

In addition to the knowledge that users create in the system via the Web, Freebase also collects much information from Wikipedia [23] including Notable Names Database (NNDB), Fashion Model Directory (FMD), and MusicBrainz, in order to create a large amount of seed data. Before closing down, Freebase accumulated 2.4 billion facts in 44 million topics.

Wikidata is a project of the Wikimedia Foundation [24]. It is an open knowledge base, allowing users to manually record facts through a system designed to be easy to use, similar to Wikipedia. One interesting concept of Wikidata is the ability to keep the facts in conflict when it is not possible to conclude which fact is more accurate [25]. The "credibility" of information in Wikidata (including Wikipedia) does not focus on the "accuracy" of information more than the "provenance" of that information. For example, the population data of Mumbai is 12.5 million people, according to the Indian Bureau of Statistics but 20.5 million people when based on UN estimates. It is not the responsibility of the Wikidata community to find out what the truth is. Wikidata uses a straightforward way to store all information along with its source. The user has to choose which one to use. Currently, Wikidata has 30 million facts about 14 million topics. It can be seen that both DBpedia and Wikidata are the conversion of Wikipedia data into structured data using different methods [26]. However, some parts of Wikidata have been converted and incorporated into DBpedia Wikidata [27] and the ProFusion dataset [28].

Cyc is an extensive knowledgebase project by Douglas B. Lenat which started in 1984 [29]. The goal is to store a large number of facts and organize them automatically. OpenCyc is a smaller version of Cyc that reduces the size of the knowledge base and is publicly available [30]. However, OpenCyc was shut down in 2017, but ResearchCyc is still open for research studies [31].

3.1.3 Transform data from other formats to RDF

RDF123 by Han et al. [32] is a tool used to convert data in spreadsheet format to RDF format. Its concept can also be used to convert the table data into RDF. A survey paper [33] of the W3C RDB2RDF Incubator Group discusses many research projects that involve converting data from relational databases to RDF. Although this research does not mention the data conversion from the generic table, it can be applied to table conversion.

There are also many W3C recommendations by CSV on the Web Working Group[4] which discusses the conversion of data in the form of record sets in CSV format to other formats such as RDF or JSON.

[4] https://www.w3.org/2013/csvw/wiki/Main_Page CSV on the Web Working Group Wiki

3.2 Research related to the conversion of table and list to other formats

There are several research works that involve converting table and list into other formats.

Yang and Luk [34, 35] discuss a thorough method for converting Web-based tables into key-value pair data and provide solutions to the problem of extracting data from the table in various cases.

The research of Pivk, Cimiano, and Sure [36] proposes a method to convert data from the Web-based table into F-logic (frame logic) which is a frame representation that can be applied to Semantic Web.

Table Analysis for Generating Ontologies (TANGO) is the research of Embley [37] and Tijerino et al. [38, 39]. The goal is to transform table data into ontology.

Table Extraction by Global Record Alignment (TEGRA) by Chu et al. [40] discusses the challenges of extracting structured data from Web-based table, in which in some case, a "table" that appear on a Webpage is not in HTML table format but it may be in HTML list or other arrangements.

DeExcelerator is the research of Eberius et al. [41]. It is a framework for extracting structured data from HTML tables and spreadsheets.

Venetis et al. [42] solve the problem of dealing with semantics and ontologies by manually adding classes to column headers of the table without having to do schema matching. However, the user must have the skill to add this information.

WebTables [43, 44] is a project of Google Research to extract structured data from HTML tables on Webpages. They searched 14.1 billion HTML tables and found that only 154 million tables have sufficient quality to allow extraction of the structured data [45]. Most HTML tables on the web are used to define the layout of the webpage but are not used to present the data in the actual table format [46]. WebTables uses the classifier that is adjusted to focus on recall more than precision in order to filter the table from the Webpage as much as possible. It then selects only the table with a single-line header and ignores other more complicated tables. Later, this project has been developed into a system called Octopus [47] to help support the search engine more efficiently.

At Google, Elmeleegy et al. [48] use WebTables to support a system called ListExtract to extract 100,000 lists from the web and then transform them into relational databases. Wong et al. [49] use 1000 machines to extract 10.1 billion tuples from 1 billion Webpages with parallel algorithms in less than 6 hours.

Fusion Tables [50] is a Google Research project designed to allow users to upload table data on the web for data analysis with various tools. It is currently available on Google Docs.

The Web Data Commons (WDC) [51] is a project to extract structured data from Common Crawl which is the largest webpage archive that is publicly available. A part of the WDC called Web Table Corpora only extracts structured data from HTML tables in the Common Crawl Web archive. Currently, Web Table Corpora has been available to download in two sets. The first set is the 2012 Corpus which extracts 147 million tables from 3.5 billion Webpages in 2012 Common Crawl. The second set is the 2015 Corpus which extracted 233 million tables from 1.78 billion webpages in July 2015 Common Crawl. The second set contains metadata about extracted tables, while this information is not reserved in the first set.

WDC Web Table Corpus has been used in many research. For example, it is used to measure the performance of schema matching approaches for various levels of table elements (such as table-to-class, row-to-instance, and attribute-to-property

[4] https://www.w3.org/2013/csvw/wiki/Main_Page CSV on the Web Working Group Wiki

matching) which previously used different datasets thus making it difficult to be compared [52].

The most similar work to our proposal is WikiTables [53] which is a tool to extract information from the tables in Wikipedia. It is used to discover new hidden facts. The result of this research is a set of 15 million tuples extracted from the Wikipedia tables.

3.3 Current "standard" representation of table and list

There are many ways to represent tables and lists in the standard data formats issued by many standard bodies such as:

- International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) standards
- Internet Engineering Task Force (IETF) Request for Comments (RFC)
- World Wide Web Consortium (W3C) Recommendations (REC)
- Internet Assigned Numbers Authority (IANA) Multipurpose Internet Mail Extensions (MIME) media types

We mention only the most common standard and format that are capable of table and list representation such as:

- Comma-separated values (CSV), i.e., RFC 4180 [54], and other delimiter-separated values (DSV) such as tab-separated values (TSV)[5].
- Markup languages such as HTML table, i.e., RFC 1942 [55] (developed from USDOD SGML Table Model), and HTML list in HTML 2.0 RFC 1866 [56].
- Lightweight markup languages such as Wikitext table and list and other markdown languages. After many attempts to standardize various of them, they end up with RFC 7763 for the original syntax and RFC 7764 for other variants.
- Office spreadsheet and word processor table/list, e.g., OASIS OpenDocument Format (ODF) ISO/IEC 26300 and Microsoft Office Open XML (OOXML) ISO/IEC 29500. ISO/IEC also issues a comparison of both formats and guidelines for translation between them in ISO/IEC TR 29166.

4. TULIP: table/list interchangeable, unified, pivotal vocabulary

The main idea of TULIP is to transform the semi-structured data in the form of tables and lists, regardless of the source, to the structured data in the form of five-star open data as a set of RDF triples. Each triple contains only subject-predicate-object. The triples are connected to other triples and form a directed graph called the RDF graph. That is the principle of Linked Data, allowing Semantic Web

[5] https://www.iana.org/assignments/media-types/text/tab-separated-values

applications to consume TULIP's five-star open data in the same way as another Linked Data.

4.1 TULIP in a *bud* shell

TULIP is a set of RDF vocabulary (the completed TULIP specification is available at http://purl.org/tulip/spec) in the form of RDF Schema. It consists of a set of RDF properties and RDF classes that are used to define structures for data representation to completely store table and list data and preserve all of its original semantic structure. It includes basic properties needed for five-star open data such as identifiable, dereferenceable, etc., as well as the three unique properties of TULIP: interchangeable, unified, and pivotal.

4.1.1 TULIP interchangeable property

TULIP has the complete preservation property in order to preserve the semantic structure of the source table and list, such as the cell contents, table structure, column/row headers, list items, and hierarchy including some formats such as spanning cells, but not including decorative style, for example, typographic styles (bold/italics), fonts, colors, borders, backgrounds, etc. The original tables and lists can be recreated from the RDF triple set using TULIP vocabulary.

This is possible because TULIP has a set of properties and classes to store the content data and classes, such as tables, columns, rows, table cells, lists, list items, etc., as well as various characteristics such as table headers, spanning cells, enumerated lists, etc.

4.1.2 TULIP unified property

TULIP retains both the tables and lists in the same format as the hierarchical treelike structure. The structure is stored as a set of RDF triples which can represent directed graphs without order or precedence between sibling nodes. So, a set of additional RDF properties must be defined to mimic the hierarchical structures and precedence of nodes in the hierarchy. That resembles a feature of the RDF called the RDF Container; however, the TULIP has more specific features.

As TULIP retains the structure in this way, the output from a query can be projected to a new structure different from the original. It depends on how an application wants to present the information. TULIP data is stored with standard RDF properties such as rdf:type and rdfs:label. There are special RDF properties to model the treelike structure. The structure is overlaid on top of the standard RDF graph. So, Semantic Web applications that do not understand the TULIP schema can perform graph traversal and get all the contents from TULIP.

The advantage is that we can look at the data without paying attention to the origin of what type of data it came from. Instead, we can choose to look at it the way we want. For example, we can look at the data that originated from a table but think of it as if it comes from a list or vice versa. Otherwise, we may combine data in both formats. It is possible to show the data in entirely different formats, such as charts or diagrams. Therefore, if we want to create an infographic from TULIP data, we can create it dynamically and change its appearance freely in any form.

4.1.3 TULIP pivotal property

Pivotal properties or view manipulation can be done because TULIP has another type of structure modeled to mimic the multidimensional array on the RDF graph.

It can store data both in column-orient (columnar) and row-orient (row-based). This allows us to query specific content of all sizes and dimensions in a single query.

Moreover, with this model, we can apply the principles of data warehouse and online analytical processing (OLAP) operations, such as rolling up the entire data in the same group, drilling down to any layer, or even slicing to cut only some axes of multidimensional content including dicing, i.e., rotate to change the perspective which means that we can filter and pivot the view of the data in TULIP format any way we want.

One of the key concepts of TULIP is using an RDF feature called RDF collection, i.e., RDF list. Apply it as a one-dimensional array to store subscripts of each level in a multidimensional array by placing all subscripts as corresponding members of the RDF collection. Then put these collections to each node of TULIP. Access to each element of TULIP can be done by a SPARQL querying for its RDF collection items that match to the corresponding subscripts.

4.2 Creating five-star open data table and list with TULIP schema

Now, we will demonstrate how to create RDF triples using the TULIP schema to represent a simple table. We use the following small example table of three columns by three rows.

Cell Content 1,1	Cell Content 2,1	Cell Content 3,1
Cell Content 1,2	Cell Content 2,2	Cell Content 3,2
Cell Content 1,3	Cell Content 2,3	Cell Content 3,3

Because TULIP schema can represent the table in both column-oriented (column-major) or row-oriented (row-major), in this case, we represent a table with a column-major format. The sample data in the table cells is preceded by the corresponding column number, followed by the row number.

Excerpts of RDF triples used to represent the three-column by three-row table above using TULIP schema are shown in **Figure 1**.

```
ex:TableExample
    tlp:member _:Table1 .
_:Table1 rdf:type tlp:Table ;
    tlp:index 1 ;
    tlp:member _:Col1, _:Col2, _:Col3 .
_:Col1 rdf:type tlp:Column ;
    tlp:index 1 ;
    tlp:member _:Cell11, _:Cell12, _:Cell13 .
_:Cell11 rdf:type tlp:Cell ;
    tlp:index 1 ;
    rdfs:label "Cell Content 1,1" .
_:Cell12 rdf:type tlp:Cell ;
    tlp:index 2 ;
    rdfs:label "Cell Content 1,2" .
. . .
_:Cell33 rdf:type tlp:Cell ;
    tlp:index 3 ;
    rdfs:label "Cell Content 3,3" .
```

Figure 1.
RDF triples of the example table represented by the TULIP schema.

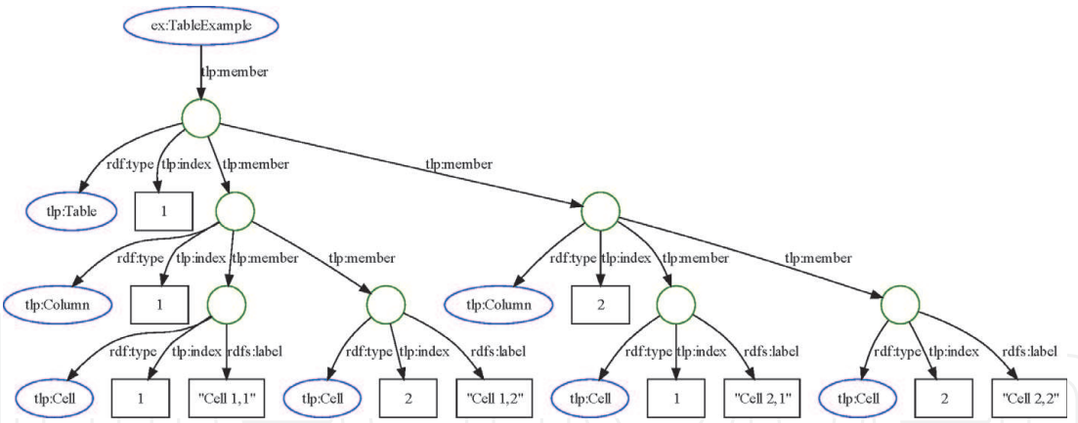

Figure 2.
RDF graph of the example table using TULIP schema.

Figure 2 shows these RDF triples in the RDF graph. (To make the figure more compact, we adjusted the table to a two-column by two-row dimension.)

Next, we will demonstrate how to create a simple five-star open data list using the TULIP schema by using the following example list.

- List Item 1
- List Item 2
    - List Item 2,1
    - List Item 2,2
        i. List Item 2,2,1
- List Item 3

The RDF triples representing the above list using the TULIP schema are shown in **Figure 3**. The RDF graph of these RDF triples is shown in **Figure 4**.

```
ex:ListExample
      tlp:member _:List1 .
_:List1 rdf:type tlp:List ;
      tlp:index 1 ;
      tlp:member _:Item1, _:Item2, _:Item3 .
_:Item1 rdf:type tlp:Item ;
      tlp:index 1 ;
      rdfs:label "List Item 1" .
_:Item2 rdf:type tlp:Item ;
      tlp:index 2 ;
      rdfs:label "List Item 2" ;
      tlp:member _:Item21, _:Item22 .
. . .
_:Item22 rdf:type tlp:Item ;
      tlp:index 2 ;
      rdfs:label "List Item 2,2" ;
      tlp:member _:Item221 .
_:Item221 rdf:type tlp:Item ;
      tlp:index 1 ;
      rdfs:label "List Item 2,2,1".
```

```
_:Item3 rdf:type tlp:Item ;
    tlp:index 3 ;
    rdfs:label "List Item 3" .
```

Figure 3.
RDF triples of the example list represented by the TULIP schema.

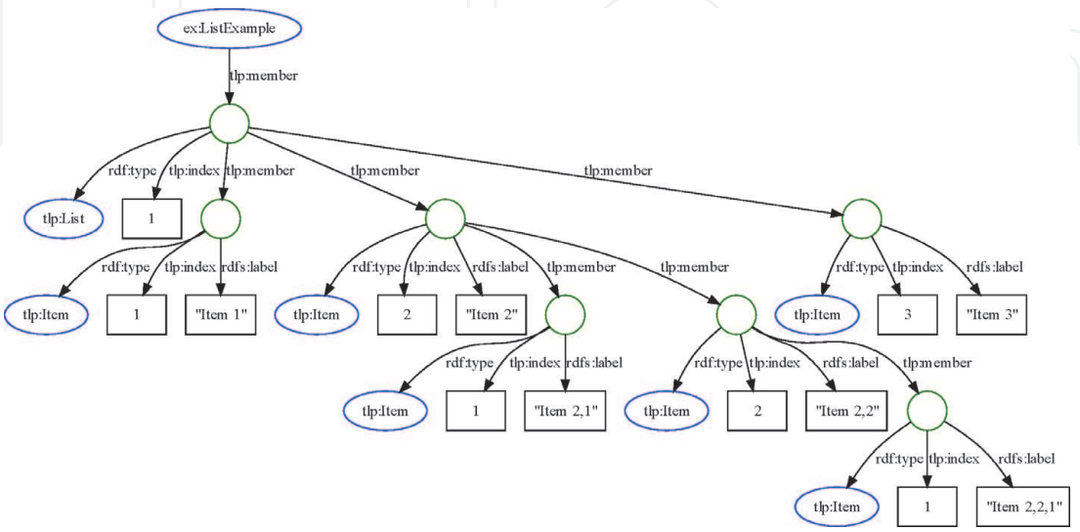

Figure 4.
RDF graph of the example list using TULIP schema.

4.3 Providing the way for direct data access

The RDF graphs, both in **Figures 2** and **4**, are already in a hierarchical structure and sequence with tlp:index (hereafter referred to as the "TULIP indexed member, TIM model"). It can be used to recreate the original table and list. This can be achieved by graph traversal. Furthermore, generic Semantic Web applications can access the data hierarchically. This is not much different from standard RDF containers. To access each arbitrary data, we have to indirectly access by performing graph traversal step by step until we reach the data we need. Writing SPARQL queries to perform this task is not easy. Therefore, we provide direct data access by assigning "position" to each data element. Similar to creating a multidimensional array index, we mimic this concept by using the feature of RDF called collections, i.e., RDF lists, to provide a way to access data directly in a single query without any nested match (hereafter referred to as the "TULIP index list, TIL model"). We use the tlp: element to point to the blank node of each item in the flat structure and create the tlp: indexList property for each node. The tlp:indexList property has its range, i.e., object in rdf:List class of sequence number according to the tlp:index in each node of the TIM model, and ends with zero (used to specify whether to separate only single element or group of all elements under the same parent node, more about this will be discussed later). If we take the example table and list to create the RDF triples using the TIL model, they will look like **Figure 5** and **Figure 6**.

```
ex:TableExample
    tlp:element _:Table1,
                _:Col1, _:Cell11, _:Cell12, _:Cell13,
```

```
                    _:Col2, _:Cell21, _:Cell22, _:Cell23,
                    _:Col3, _:Cell31, _:Cell32, _:Cell33 .
_:Table1 rdf:type tlp:Table ;
    tlp:indexList ( 1 0 ) .
_:Col1 rdf:type tlp:Column ;
    tlp:indexList ( 1 1 0 ) .
_:Cell11 rdf:type tlp:Cell ;
    tlp:indexList ( 1 1 1 0 ) ;
    rdfs:label "Cell Content 1,1" .
_:Cell12 rdf:type tlp:Cell ;
    tlp:indexList ( 1 1 2 0 ) ;
    rdfs:label "Cell Content 1,2" .
. . .
_:Cell33 rdf:type tlp:Cell ;
    tlp:indexList ( 1 3 3 0 ) ;
    rdfs:label "Cell Content 3,3".
```

Figure 5.
RDF triples of the example table represented by TIL model.

```
ex:ListExample
    tlp:element _:List1,
                _:Item1,
                _:Item2, _:Item21, _:Item22, _:Item221,
                _:Item3 .
_:List1 rdf:type tlp:List ;
    tlp:indexList ( 1 0 ) .
_:Item1 rdf:type tlp:Item ;
    tlp:indexList ( 1 1 0 ) ;
    rdfs:label "List Item 1" .
_:Item2 rdf:type tlp:Item ;
    tlp:indexList ( 1 2 0 ) ;
    rdfs:label "List Item 2" .
. . .
_:Item22 rdf:type tlp:Item ;
    tlp:indexList ( 1 2 2 0 ) ;
    rdfs:label "List Item 2,2" .
_:Item221 rdf:type tlp:Item ;
    tlp:indexList ( 1 2 2 1 0 ) ;
    rdfs:label "List Item 2,2,1" .
_:Item3 rdf:type tlp:Item ;
    tlp:indexList ( 1 3 0 ) ;
    rdfs:label "List Item 3" .
```

Figure 6.
RDF triples of the example list represented by TIL model.

An example of the RDF graph for **Figure 6** (shows only the first five nodes) is in **Figure 7**.

Each of the tlp:indexList objects is the structure of the RDF collection, i.e., RDF list, which when extended, becomes an unbalanced binary tree structure, where leaf nodes are each member of the list, respectively. For example, the tlp:indexList (1 2 0) has a structure as **Figure 8**.

Access to each element of TULIP by SPARQL is achieved by querying its tlp: indexList that have RDF collection items matching the corresponding subscripts. The problem is that the current SPARQL specification has limited ability to directly handle the RDF collection (causing many attempts to create the alternatives to the

DOI: http://dx.doi.org/10.5772/intechopen.91406

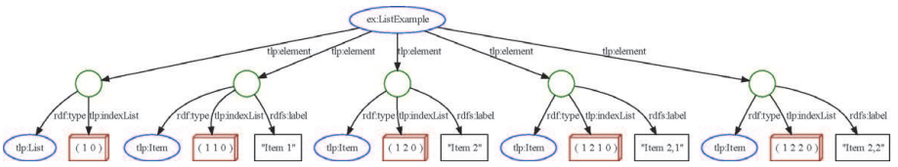

Figure 7.
RDF graph of the example list represented by TIL model.

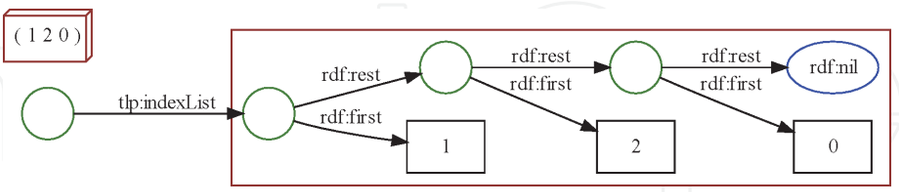

Figure 8.
RDF graph of tlp:indexList (1 2 0) in detailed structure.

standard RDF collection such as ordered list ontology [57] and collection ontology [58] or ideas to improve SPARQL to access RDF collections more conveniently, such as the proposal of Leigh and Wood [59]). A temporary solution to the problem, while waiting for the next SPARQL specification that might have improvements in this matter, is to use a feature called Property Path provided in SPARQL 1.1 to access each RDF collection item.

For example, if we want the data in the second item of the first list in the previous example, we can query for the elements that have the tlp:indexList matching to (1 2) by SPARQL query as follows:

```
SELECT ?label
WHERE {
        ex:ListExample tlp:element ?elem .
        ?elem tlp:indexList/rdf:first 1 .
        ?elem tlp:indexList/rdf:rest/rdf:first 2 .
        ?elem rdfs:label ?label .
}
        The results are as follows:
"List Item 2"
"List Item 2,1"
"List Item 2,2"
"List Item 2,2,1"
```

These are all items that are under the second list item. In case if an only a main item is needed, we could add the last subscript by zero to match only one node. That can be done by just adding one more line of SPARQL query to match the tlp: indexList (1 2 0). That is:

```
SELECT ?label
WHERE {
        ex:ListExample tlp:element ?elem .
        ?elem tlp:indexList/rdf:first 1 .
        ?elem tlp:indexList/rdf:rest/rdf:first 2 .
        ?elem tlp:indexList/rdf:rest/rdf:rest/rdf:first 0 .
        ?elem rdfs:label ?label .
}
The result is:
"List Item 2"
```

Similar to the list example, if we want the entire third column of the first table, we match the tlp:indexList with (1 3) as follows:

```
SELECT ?label
WHERE {
        ex:TableExample tlp:element ?elem .
        ?elem tlp:indexList/rdf:first 1 .
        ?elem tlp:indexList/rdf:rest/rdf:first 3 .
        ?elem rdfs:label ?label .
}
```

The results are as follows:

"Cell Content 3,1"
"Cell Content 3,2"
"Cell Content 3,3"

Alternatively, if we want the whole third row, we match the tlp:indexList with (1 ? 3) where "?" at the second subscript position is the tlp:Column level, which we will not filter. So we will get every column.

```
SELECT ?label
WHERE {
        ex:TableExample tlp:element ?elem .
        ?elem tlp:indexList/rdf:first 1 .
        ?elem tlp:indexList/rdf:rest/rdf:rest/rdf:first 3 .
        ?elem rdfs:label ?label .
}
```

The results are:

"Cell Content 1,3"
"Cell Content 2,3"
"Cell Content 3,3"

4.4 Combining the advantages of both models

Both TULIP models, indexed member model and index list model, have different pros and cons. Users can choose either type as appropriate to their needs. Furthermore, they could create RDF triples using the merged model called the "TULIP hybrid, TH model," which combines the advantages of both types in one structure. The idea is that we take the TIM model as the basis and then add TIL elements by inserting a tlp:indexList property into each blank node of TIM and adding all tlp:element to the primary resource. For example, when adding tlp: indexList and tlp:element to **Figure 1**, it will look like **Figure 9**.

```
ex:TableExample
    tlp:element _:Table1,
                _:Col1, _:Cell11, _:Cell12, _:Cell13,
                _:Col2, _:Cell21, _:Cell22, _:Cell23,
                _:Col3, _:Cell31, _:Cell32, _:Cell33 ;
    tlp:member _:Table1 .
_:Table1 rdf:type tlp:Table ;
    tlp:index 1 ;
    tlp:indexList ( 1 0 ) ;
    tlp:member _:Col1, _:Col2, _:Col3 .
. . .
_:Cell33 rdf:type tlp:Cell ;
    tlp:index 3 ;
    tlp:indexList ( 1 3 3 0 ) ;
    rdfs:label "Cell Content 3,3" .
```

Figure 9.
RDF triples of the example table represented by TH model.

DOI: http://dx.doi.org/10.5772/intechopen.91406

When shown as RDF graph, it will look like **Figure 10**. (To make the graph more compact, we have resized the table to two columns by two rows and remove some nodes, and many edges of tlp:element have also been omitted. In fact, tlp:element will point to every blank node).

Likewise, the TH model RDF triples of the example list are in **Figure 11**. When shown as the RDF graph, it will look like **Figure 12** (showing only some tlp:element edges to make the graph more convenient and clear).

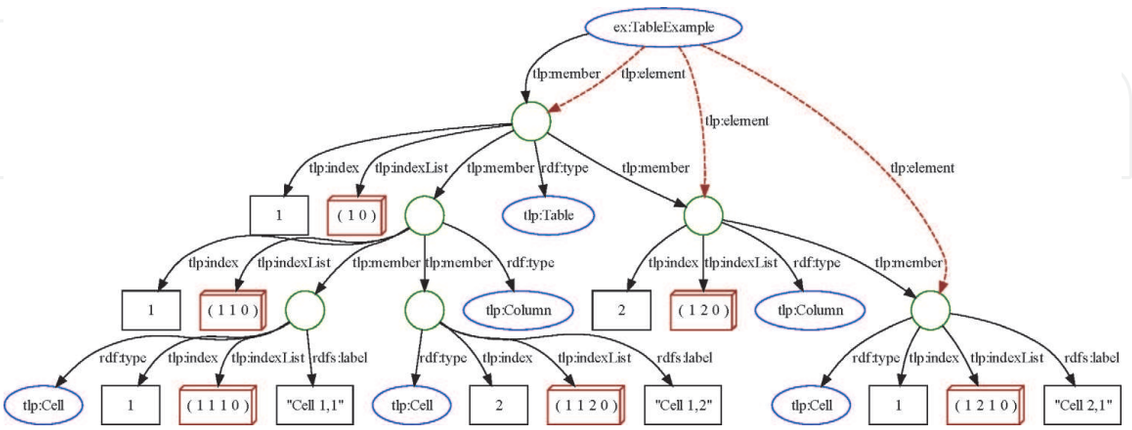

Figure 10.
RDF graph of the example table represented by TH model.

```
ex:ListExample
    tlp:element _:List1,
                _:Item1,
                _:Item2, _:Item21, _:Item22, _:Item221,
                _:Item3 ;
    tlp:member _:List1 .
_:List1 rdf:type tlp:List ;
    tlp:index 1 ;
    tlp:indexList ( 1 0 ) ;
    tlp:member _:Item1, _:Item2, _:Item3 .
. . .
_:Item3 rdf:type tlp:Item ;
    tlp:index 3 ;
    tlp:indexList ( 1 3 0 ) ;
    rdfs:label "List Item 3" .
```

Figure 11.
RDF triples of the example list represented by TH model.

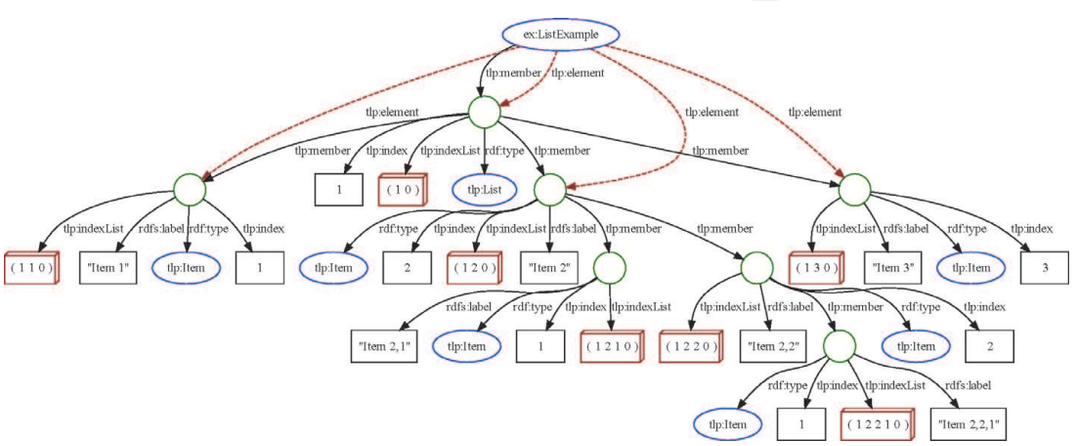

Figure 12.
RDF graph of the example list represented by TH model.

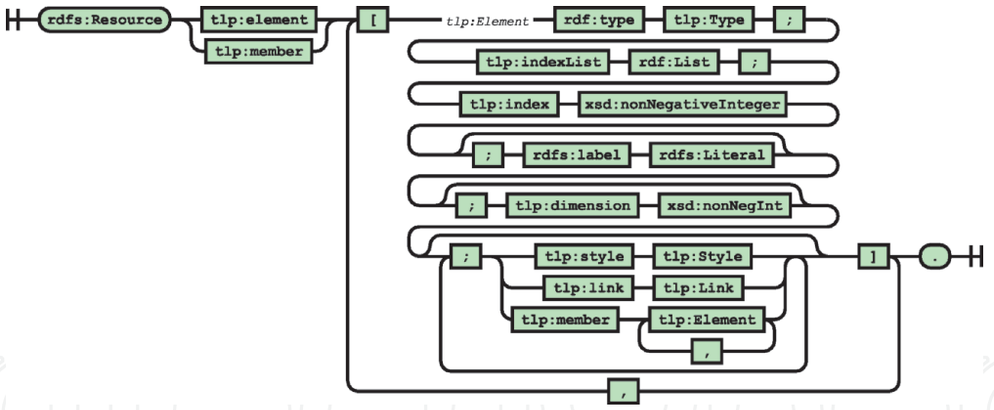

Figure 13.
Excerpt of TULIP vocabulary syntaxes tlp:element and tlp:member.

An excerpt of TULIP vocabulary syntax (detail specification can be found at http://purl.org/tulip/spec) for tlp:element and tlp:member is shown in **Figure 13**.

5. Experimental results and contributions

We have designed and implemented two reference libraries. The first library is a Python library that has functions to extract/transform Webpages and create TULIP datasets. The second library is a JavaScript library to query TULIP endpoint, consume its result sets, and manipulate them. The code is provided in the GitHub repositories at https://github.com/julthep/tulip and https://github.com/julthep/tulip.js respectively.

Furthermore, we experimented with TULIP vocabulary using Wikipedia as the data source by converting a number of articles into TULIP data format. The result datasets can be accessed via SPARQL endpoint at http://tlpedia.org/sparql/ or downloaded at http://tlpedia.org/datasets/. These datasets will be updated periodically, and the number of imported articles will be increased on a regular basis.

6. Conclusion

Our proposal is different from existing research, which mainly efforts on transforming data from tables and lists into facts in various formats. TULIP focuses on extracting data from tables and lists into the dataset in the form of five-star Linked Data. Also, the RDF triples result can be used to recreate tables and lists in the same format as the source data because the designed schema focuses on the ability to preserve the structure of the original table and list. Another essential feature is that the acquired RDF triples can also be embedded in a package file such as XML or HTML with RDFa to be used to create tables and lists on a Webpage as an integrated dataset.

TULIP can also be applied to many applications since it is designed to be very flexible. Implementers can choose to use any of its schema models, depending on their usage. It also supports many types of data and is extensible. Actually, TULIP also supports the Document Object Model (DOM) which can transform paragraphs or text blocks into the same structure. It can be used to transforms Wikipedia articles into TULIP format as a five-star open dataset so that the Semantic Web application can consume Linked Data more conveniently. All of this is to create a data structure that is not just machine-readable but will be machine-understandable.

DOI: http://dx.doi.org/10.5772/intechopen.91406

Author details

Julthep Nandakwang* and Prabhas Chongstitvatana
Department of Computer Engineering, Chulalongkorn University, Bangkok, Thailand

*Address all correspondence to: julthep@nandakwang.com

References

[1] Spivack N. Web 3.0–The Best Official Definition Imaginable [Internet]. 2007. Available from: http://www.novaspivack.com/technology/web-3-0-the-best-official-definition-imaginable [Accessed: 20 December 2019]

[2] Berners-Lee T, Cailliau R, Luotonen A, Nielsen HF, Secret A. The world-wide web. Communications of the ACM. 1994;**37**(8):76-82

[3] Berners-Lee T, Hendler J, Lassila O. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American. 1 May 2001;**284**(5):34-43

[4] Bratt S. Semantic Web and Other W3C Technologies to Watch [Internet]. W3C; 2006. Available from: https://www.w3.org/2006/Talks/1023-sb-W3CTechSemWeb [Accessed: 20 December 2019]

[5] Shadbolt N, Hall W, Berners-Lee T. The semantic web revisited. Intelligent Systems, IEEE. 2006;**21**(3):96-101

[6] Bizer C, Heath T, Berners-Lee T. Linked Data - the Story So Far. Semantic Services. Interoperability and Web Applications: Emerging Concepts. 2009: 205-227

[7] Bizer C. The emerging web of linked data. IEEE Intelligent Systems. 2009; **24**(5)

[8] Färber M, Ell B, Menne C, Rettinger A. A comparative survey of DBpedia, freebase, OpenCyc, Wikidata, and YAGO. Semantic Web Journal. July 2015;**1**(1):1-5

[9] Ringler D, Paulheim H. One knowledge graph to rule them all? Analyzing the differences between DBpedia, YAGO, Wikidata & co. In: Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz). Cham: Springer; 25 September 2017:366-372

[10] Färber M, Bartscherer F, Menne C, Rettinger A. Linked data quality of DBpedia, freebase, OpenCyc, Wikidata, and YAGO. Semantic Web. 2018;**9**(1): 77-129

[11] Pillai SG, Soon L-K, Haw S-C. Comparing DBpedia, Wikidata, and YAGO for web information retrieval. In: Intelligent and Interactive Computing. Singapore: Springer; 2019. pp. 525-535

[12] Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes PN, et al. DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web. 1 January 2015;**6**(2):167-195

[13] Auer S, Lehmann J. What have Innsbruck and Leipzig in common? Extracting semantics from wiki content. In: Franconi E, Kifer M, May W, editors. 4th European Semantic Web Conference, ESWC 2007, June 3–7, 2007 Proceedings; 2007/01/01. Innsbruck, Austria: Springer Berlin Heidelberg; 2007. pp. 503-517

[14] Bizer C, Lehmann J, Kobilarov G, Auer S, Becker C, Cyganiak R, et al. DBpedia - a crystallization point for the web of data. Web Semantics: Science, Services and Agents on the World Wide Web. 2009;7(3):154-165

[15] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A nucleus for a web of open data. In: Aberer K, Choi K-S, Noy N, Allemang D, Lee K-I, Nixon L, et al., editors. 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, November 11-15, 2007 Proceedings; 2007/01/01. Busan, Korea: Springer Berlin Heidelberg; 2007. pp. 722-735

[16] Isbell J, Butler MH. Extracting and re-using structured data from wikis. Bristol: Digital Media Systems Laboratory of Hewlett-Packard Development Company; November 14, 2007. Report No.: HPL-2007-182

[17] Suchanek FM, Kasneci G, Weikum G. A core of semantic knowledge unifying WordNet and Wikipedia. In: YAGO, editor. Proceedings of the 16th International Conference on World Wide Web. Banff, Alberta, Canada: ACM; 2007. p. 1242667

[18] Navigli R, Ponzetto SP. BabelNet: Building a very large multilingual semantic network. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden. 1858704: Association for Computational Linguistics; 2010. pp. 216-225

[19] Gd M, Weikum GMENTA. Inducing multilingual taxonomies from Wikipedia. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. Toronto, ON, Canada. 1871577: ACM; 2010. pp. 1099-1108

[20] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. 9 June 2008. pp. 1247-1250

[21] Bollacker K, Tufts P, Pierce T, Cook R, editors. A platform for scalable, collaborative, structured information integration. In: International Workshop on Information Integration on the Web (IIWeb'07); 2007 July

[22] Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, et al. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, New York, USA. 2623623: ACM; 2014. pp. 601-610

[23] Bollacker K, Cook R, Tufts P, editors. Freebase: A shared database of structured general human knowledge. In: Proceedings of the 22nd national conference on Artificial intelligence-2. Vol 7. 22 July 2007. pp. 1962-1963

[24] Vrandečić D, Krötzsch M. Wikidata: A free collaborative knowledgebase. Communications of the ACM. 2014; **57**(10):78-85

[25] Vrandečić D. The rise of Wikidata. IEEE Intelligent Systems. 2013;**28**(4): 90-95

[26] Abián D, Guerra F, Martínez-Romanos J, Trillo-Lado R. Wikidata and DBpedia: A comparative study. In: Semanitic Keyword-Based Search on Structured Data Sources. Cham: Springer; 11 September 2017. pp. 142-154

[27] Ismayilov A, Kontokostas D, Auer S, Lehmann J, Hellmann S. Wikidata through the eyes of DBpedia. Semantic Web. 2018;**9**(4):493-503

[28] Frey J, Hofer M, Obraczka D, Lehmann J, Hellmann S. DBpedia FlexiFusion the best of Wikipedia> Wikidata> your data. In: International Semantic Web Conference. Cham: Springer; 26 October 2019. pp. 96-112

[29] Lenat DB. CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM. 1995; **38**(11):33-38

[30] Lenat DB. The voice of the turtle: Whatever happened to AI? AI Magazine. 2008;**29**(2):11

[31] Lenat DB. Building a machine smart enough to pass the Turing test. In: Epstein R, Roberts G, Beber G, editors.

Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer. Dordrecht: Springer Netherlands; 2009. pp. 261-282

[32] Han L, Finin T, Parr C, Sachs J, Joshi A. RDF123: From Spreadsheets to RDF. In: Sheth A, Staab S, Dean M, Paolucci M, Maynard D, Finin T, et al., editors. The Semantic Web - ISWC 2008. Lecture Notes in Computer Science. 5318. Berlin Heidelberg: Springer; 2008. pp. 451-466

[33] Sahoo SS, Halb W, Hellmann S, Idehen K, Thibodeau T Jr, Auer S, et al. A survey of current approaches for mapping of relational databases to RDF. W3C RDB2RDF Incubator Group Report; 2009

[34] Yang Y. Web Table Mining and Database Discovery. Doctoral Dissertation, Simon Fraser University; 2002

[35] Yang Y, Luk W-S. A framework for web table mining. In: Proceedings of the 4th International Workshop on Web Information and Data Management. McLean, Virginia, USA. 584940: ACM; 2002. pp. 36-42

[36] Pivk A, Cimiano P, Sure Y. From Tables to Frames. In: McIlraith SA, Plexousakis D, van Harmelen F, editors. The Semantic Web – ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-112004, Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004. p. 166–181.

[37] Embley DW, Tao C, Liddle SW. Automatically extracting ontologically specified data from HTML tables of unknown structure. In: Spaccapietra S, March ST, Kambayashi Y, editors. Conceptual Modeling — ER 2002: 21st International Conference on Conceptual Modeling Tampere, Finland, October 7–11, 2002 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg; 2003. pp. 322-337

[38] Tijerino YA, Embley DW, Lonsdale DW, Nagy G, editors. Ontology generation from tables. In: Proceedings of the fourth international conference on web information systems engineering, WISE; 2003. pp. 10-12

[39] Tijerino YA, Embley DW, Lonsdale DW, Ding Y, Nagy G. Towards ontology generation from tables. World Wide Web. 2005;**8**(3):261-285

[40] Chu X, He Y, Chakrabarti K, Ganjam K. TEGRA: Table extraction by global record alignment. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. Melbourne, Victoria, Australia. 2723725: ACM; 2015. pp. 1713-1728

[41] Eberius J, Werner C, Thiele M, Braunschweig K, Dannecker L, Lehner W. DeExcelerator: A framework for extracting relational data from partially structured documents. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management. San Francisco, California, USA. 2508210: ACM; 2013. pp. 2477-2480

[42] Venetis P, Halevy A, Madhavan J, Paşca M, Shen W, Wu F, et al. Recovering semantics of tables on the web. Proceedings of the VLDB Endowment. 2011;**4**(9):528-538

[43] Cafarella MJ, Halevy A, Wang DZ, Wu E, Zhang Y. WebTables: Exploring the power of tables on the web. Proc VLDB Endow. 2008;**1**(1):538-549

[44] Balakrishnan S, Halevy AY, Harb B, Lee H, Madhavan J, Rostamizadeh A, et al., editors. Applying WebTables in practice. In: CIDR. 2015

[45] Cafarella MJ, Halevy AY, Zhang Y, Wang DZ, Wu E, editors. Uncovering the Relational Web. WebDB; June 13, 2008

[46] Wang Y, Hu J, editors. A machine learning based approach for table detection on the web. In: Proceedings of the 11th International Conference on World Wide Web; 2002 May. Honolulu, Hawaii, USA: ACM; 2002. p. 511478

[47] Cafarella MJ, Halevy A, Khoussainova N. Data integration for the relational web. Proc VLDB Endow. 2009;**2**(1):1090-1101

[48] Elmeleegy H, Madhavan J, Halevy A. Harvesting relational tables from lists on the web. Proc VLDB Endow. 2009;**2**(1):1078-1089

[49] Wong YW, Widdows D, Lokovic T, Nigam K. Scalable attribute-value extraction from semi-structured text. In: IEEE International Conference on Data mining workshops, 2009 ICDMW'09. IEEE; 6 December 2009. pp. 302-307

[50] Gonzalez H, Halevy A, Jensen CS, Langen A, Madhavan J, Shapley R, et al. Google fusion tables: Data management, integration and collaboration in the cloud. In: Proceedings of the 1st ACM Symposium on Cloud Computing. Indianapolis, Indiana, USA. 1807158: ACM; 2010. pp. 175-180

[51] Lehmberg O, Ritze D, Meusel R, Bizer C. A large public corpus of web tables containing time and context metadata. In: Proceedings of the 25th International Conference Companion on World Wide Web. Montréal, Québec, Canada. 2889386: International World Wide Web Conferences Steering Committee; 2016. pp. 75-76

[52] Ritze D, Bizer C. Matching web tables to DBpedia-a feature utility study. Context. 2017;**42**(41):19-31

[53] Bhagavatula CS, Noraset T, Downey D. Methods for exploring and mining tables on Wikipedia. In: Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics. Chicago, Illinois. 2501516: ACM; 2013. pp. 18-26

[54] Shafranovich Y. Common Format and MIME Type for Comma-Separated Values (CSV) Files [Internet]. IETF; 2005. Available from: https://tools.ietf.org/html/rfc4180 [Accessed: 20 December 2019]

[55] Raggett D. HTML Tables [Internet]. IETF; 1996. Available from: https://tools.ietf.org/html/rfc1942 [Accessed: 20 December 2019]

[56] Berners-Lee T, Connolly D. Hypertext Markup Language - 2.0 [Internet]. IETF; 1995. Available from: https://tools.ietf.org/html/rfc1866 [Accessed: 20 December 2019]

[57] Abdallah SA, Ferris B. The Ordered List Ontology Specification [Internet]. 2010. Available from: http://purl.org/ontology/olo/core# [Accessed: 20 December 2019]

[58] Ciccarese P, Peroni S. The collections ontology: Creating and handling collections in OWL 2 DL frameworks. Semantic Web. 2014;**5**(6):515-529

[59] Leigh J, Wood D. An ordered RDF list. W3C workshop — RDF next steps; June 26–27, 2010; National Center for Biomedical Ontology (NCBO), Stanford, Palo Alto, CA, USA: W3C; 2010