T2WML: Table To Wikidata Mapping Langauge

Pedro Szekely, Daniel Garijo, Divij Bhatia, Jiasheng Wu, Yixiang Yao and Jay Pujara
szekely@usc.edu,dgarijo@isi.edu,divijbha@usc.edu,jiashenw@usc.edu,yixiangy@isi.edu,jpujara@usc.edu
USC Information Sciences Institute
Marina del Rey, California

ABSTRACT

The web contains millions of useful spreadsheets and CSV files, but these files are difficult to use in applications because they use a wide variety of data layouts and terminology. We present Table To Wikidata Mapping Language (T2WML), a language that makes it easy to map and link arbitrary spreadsheets and CSV files to the Wikidata data model. The output of T2WML consists of Wikidata statements that can be loaded in the public Wikidata knowledge base or in a Wikidata clone repository, creating an augmented Wikidata knowledge graph that application developers can query using SPARQL.[1]

CCS CONCEPTS

• **Information systems → Extraction, transformation and loading**.

KEYWORDS

Knowledge Graphs; RDF; Entity Linking; Wikidata

ACM Reference Format:

Pedro Szekely, Daniel Garijo, Divij Bhatia, Jiasheng Wu, Yixiang Yao and Jay Pujara. 2019. T2WML: Table To Wikidata Mapping Langauge. In *Proceedings of the 10th International Conference on Knowledge Capture (K-CAP '19), November 19–21, 2019, Marina Del Rey, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3360901.3364448

1 INTRODUCTION

The web contains millions of useful spreadsheets and CSV files, including data from many government and international organizations. Most institutions offer their data in web sites where users can download the data in Excel and CSV formats. The downloaded data is seldom directly usable because, unlike databases (which use one column per variable), spreadsheets often arrange the data in different layouts.

Fig. 1 illustrates the problem using data downloaded from the United Nations web site[2] about homicide rates in different countries. We truncated and colored the files for ease of presentation. The cells with the homicide numbers are highlighted in green, the cells that provide contextual information for the value are highlighted in blue, and header cells are highlighted in dark blue. Fig. 1a shows the layout of the data provided in the UN website; Fig. 1b shows a more compact representation of the data using multi-level headers; Fig. 1c shows the layout that could be used to store the data in a database, and that can be used directly in tools such as Pandas; and Fig. 1d illustrates a common convention for arranging data by topic, by creating stacked tables that share common headings. All tables present the same homicide data. In this dataset, the interpretation of each value is defined by four cells (country, year, population and source) that identify the context for a value. In each table, the context cells are located in different parts of the table. Only in Fig. 1c (Database) the context cells are in the same row as the value; in the other tables, context cells appear in different rows, in header rows (examples a and b), or in visually distinct rows within the table (example d).

Existing languages and systems for mapping structured data to RDF, including R2RML,[3] RML [1], Karma [3] and CSV2RDF [2] are row based, designed for database layouts (Fig. 1c). The interpreters for these languages process one row at a time, using the column headers to define the properties used for generating RDF (RML supports other formats such as JSON and XML). These languages also provide preprocessing operations to define new columns, to join multiple tables and to fold/unfold columns and rows to rearrange data for row-based processing. The preprocessing transformations to convert the data in tables a), b) and d) to the database layout are beyond the processing capabilities of existing RDF mapping systems, and external scripting would be required to map these files to an ontology to generate RDF.

T2WML is a mapping language designed to meet three objectives: **(1)** Map data in arbitrary data layouts used in Excel and CSV files without the need of complex preprocessing steps to transform tables into a canonical "Database" representation. **(2)** Enable users who are not familiar with RDF to map spreadsheets and CSV files to knowledge graphs so that they can augment knowledge graphs with data useful in downstream applications. **(3)** Integrate mapping and entity linking so that the resulting output is linked to a reference knowledge graph, avoiding the need for a separate linking process that is often neglected in the current transformation workflows.

T2WML is designed for the Wikidata data model [4]. The main building block in this model is a *statement*, which consists of a subject, a predicate, an object, qualifiers and references. The subject, predicate an object part mirror their RDF counter parts. The qualifiers are predicate/object pairs that provide context information about a subject/predicate/object triple. For example, qualifiers can be used to qualify an employment relation between a person and an organization to record the period of time when the person was

[1]This material is based upon work supported by United States Air Force under Contract No. FA8650-17-C-7715.

[2]https://dataunodc.un.org/crime/intentional-homicide-victims

[3]https://www.w3.org/2001/sw/wiki/R2RML

K-CAP '19, November 19–21, 2019, Marina Del Rey, CA, USA

ACM ISBN 978-1-4503-7008-0/19/11…$15.00
https://doi.org/10.1145/3360901.3364448

a) Original

Country	Source		2000	2001	2002
Burundi	SDG	Females	1	2	2
Burundi	SDG	Males	4	5	6
Comoros	GHD Estimate	Females	2	1	
Comoros	GHD Estimate	Males	4	-	8
Djibouti	GHD Estimate	Females	2	1	3
Djibouti	GHD Estimate	Males	1	1	

b) Compact

Country	Source	Males			Females		
		2000	2001	2002	2000	2001	2002
Burundi	SDG	4	5	6	1	2	2
Comoros	GHD Estimate	4	-	8	2	1	
Djibouti	GHD Estimate	1	1		2	1	3

c) Database

Country	Source	Population	Year	Homicides
Burundi	SDG	Females	2000	1
Burundi	SDG	Females	2001	2
Burundi	SDG	Females	2002	2
Burundi	SDG	Males	2000	3
Burundi	SDG	Males	2001	4
Burundi	SDG	Males	2002	6
Comoros	GHD Estimate	Females	2000	2
Comoros	GHD Estimate	Females	2001	1
Comoros	GHD Estimate	Females	2002	

d) By year

Country	Source	Population	Homicides
2000			
Burundi	SDG	Females	1
	SDG	Males	3
Comoros	GHD Estimate	Females	2
	GHD Estimate	Males	4
Djibouti	GHD Estimate	Females	2
	GHD Estimate	Males	1
2001			
Burundi	SDG	Females	2
	SDG	Males	3
Comoros	GHD Estimate	Females	1
	GHD Estimate	Males	

Figure 1: Intentional Homicide Data (Excel file downloaded from dataunodc.un.org)

employed at the organization, or the title held. The references part is used to record sources from where the knowledge was derived so that each statement can be traced back to sources.

In the next sections we introduce the design concepts in T2WML, then introduce the language specification and system, and finally discuss our experience using it to extend Wikidata with data in spreadsheets downloaded from multiple sites.

2 T2WML DESIGN

T2WML is a template based language that models spreadsheets and CSV files using two concepts. **Data blocks**: possibly non-contiguous blocks of cells containing values of interest (green cells in Fig. 1). These blocks can occur anywhere in a table. **Context**: cells that describe the values in data blocks (blue cells in Fig 1).

The meaning of a value in a data block is defined using the values in selected context cells. For example, the value highlighted in Fig. 1a represents the number of homicides of males in Burundi in the year 2000 as reported by SDG. The context cells are spatially related to the target value cell, but in each table the spatial arrangement is different. For example, in Fig. 1a the country, source and population are in the same row, and the year is in the top row above the homicides number.

T2WML defines an expressive, yet simple language to identify the context cells for a target value cell. This cell-centric model gives T2WML the expressive power lacking in row-centric models in the languages derived from R2RML. In R2RML, a mapping statement can refer to cells in the same row, while T2WML can refer to cells in distant rows. For example, in Fig. 1b the population indicators (male, female) are in the first row, and the years are in the second row. In Fig. 1d the situation is more complex as the year cells are not identified by absolute row coordinates, but by their geometric arrangement relative to the target cell (i.e., in the nearest row above the target cell containing a value in the first column and empty values in the rest of the columns).

A second difference with existing mapping tools is that T2WML adopts a "link first" strategy where cells that identify entities in the target knowledge graph (Wikidata in our implementation) must be linked to the corresponding entities *before* the input data is processed to produce statements in the knowledge graph. The link-first strategy offers multiple benefits over the traditional link-later strategy used in existing tools. It ensures that linking is done, resulting in higher quality data, and at the same time avoids the need for users to design a URI scheme. Cells can also be linked to property nodes in Wikidata, enabling T2WML to also contextualize the property used to describe a value.

T2WML integrates a cell linking service (a cell *wikifier*). Users can submit cell ranges for *wikification* and curate the results within the T2MWL user interface (Wikifier panel in Fig. 2). In addition, T2WML is interoperable with the tools submitted to the 2019 ISWC Cell Annotation Challenge[4]).

A third difference with existing mapping tools is that T2WML uses YAML[5] instead of RDF to represent mappings. Our choice of YAML is deliberate as YAML has become a popular language among developers for representing structured data. It is easy to edit and easy to generate from software, enabling T2WML to be easily incorporated in larger software pipelines. The T2WML approach, is RDF friendly, as the processor can generate Wikidata statements in RDF, but is also accessible to developers and users unfamiliar with RDF (statements may also be generated in JSON-LD).

3 T2WML LANGUAGE

In this section we describe the most important features of the T2WML language. The formal specification of the full language is available in https://github.com/usc-isi-i2/t2wml.

Definitions. The key concept in T2WML is a cell, identified using two variables, `$col` and `$row`. `$col` ranges from column `A` until the last non-empty column, and `$row` is defined similarly. All constructs in the language are defined as functions of `$col` and `$row`.

Definition: (row and column expressions) A row expression is a function $f(\$col, \$row)$ whose value identifies a row in a table.

[4]https://www.aicrowd.com/challenges/iswc-2019-cell-entity-annotation-cea-challenge features 8 tools that achieve F-scores of over 85%.
[5]https://yaml.org/spec/1.2/spec.html

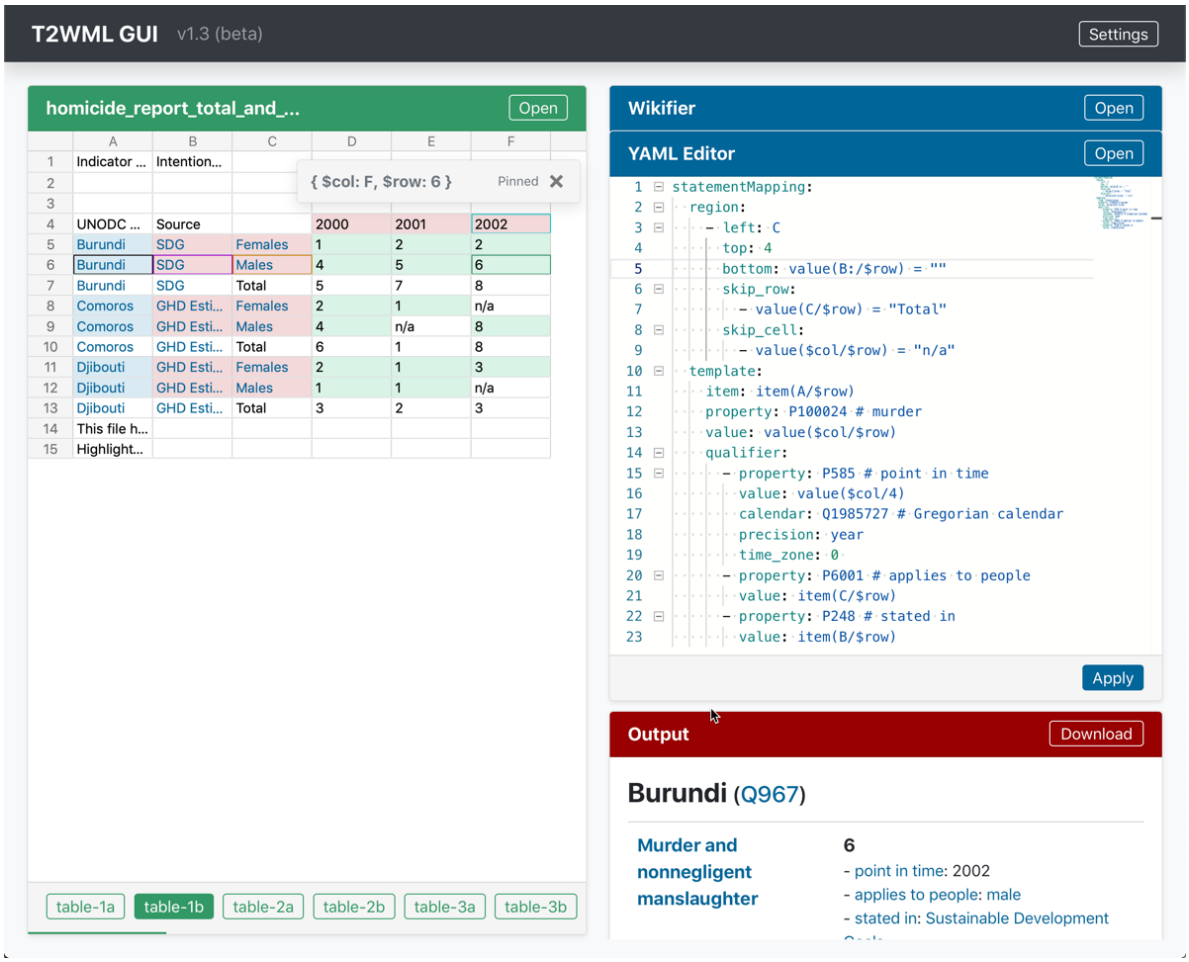

Figure 2: T2WML user interface to map the spreadsheet in Fig. 1b to Wikidata

Simple expressions include constants such as `3`, which identifies the third row regardless of the values of `$col` and `$row`, and arithmetic expressions such as `$row-2`, which identifies the row two rows above the current one (e.g., if `$row = 3` then `$row-2` designates row 1).

Complex expressions can be defined using the `->` operator which takes two arguments, a Boolean expression and A simple expression. For example, `value(A/$row) = "hello" -> $row+1` produces the row below if the cell in the current row in column `A` contains the value `hello`. If the Boolean expression is false the value is undefined. The second argument is optional allowing for a terse syntax. For example, `value(A/$row) = "hello"` is a shortcut for `value(A/$row) = "hello" -> $row` if the expression is used in a context where the value is a row.

Definition: (cell expression) A function f(column expression, row expression) whose value identifies a cell in a table. The `/` (slash) operator defines cell expressions, combining a column expression and a row expression. For example, `C/$row-2` identifies the cell in column `C` two rows above the current row. The `->` is overloaded and can be used to construct cell expressions, in addition to column or row expressions.

Definition: (value(cell expression)) Retrieves the value of a cell in a table. Fig. 2 shows examples of value expressions in lines 5, 7, 9, 13 and 16. For example, value(`$col/$row`) (line 9) identifies the value of the current cell.

Definition: (item(cell expression)) Retrieves the identifier of the item linked to the cell identified using the cell expression. The T2WML system provides multiple ways for users to define the behavior of the `item()` function. Users can invoke any wikifier service that returns results as annotations on cells, and they can also curate or define the mapping of cells to identifiers in the user interface. Fig. 2 shows examples of item expressions in lines 11, 21 and 23.

Definition: (iteration expression) Defines a sequence of cell references to iterate over until a Boolean expression returns true. Iteration expressions are defined using the `-n` or `+n` operators. The iteration starts with `n=0` and increments by one until the containing Boolean expression returns true or the cell reference containing it abandons the confines of a table. For example, in the table in Fig. 1d, the expression `value(A/$row-n)` can be used to retrieve the country that defines the context of any of the homicide counts.

3.1 T2WML Statement Mappings

We describe the language for mapping tables to statements using Fig. 2, which shows the user interface for the T2WML system. The panel on the left shows the table in Fig. 1a, and the YAML Editor panel shows a T2WML specification to map this data to Wikidata.

T2WML specifications consist of a data **`region`** to identify the cells containing the values of statements (e.g., homicide counts, highlighted in green in the Table Viewer in Fig. 2), and a **`template`** section to define the statements for each cell in the data region.

T2WML defines data regions using column and row expressions. Data regions confine an iterator to visit all cells containing data values. In our example, the data region is defined by columns `C` and right edge of the table (the default) and rows `4` and `14`. Using constant row and cell expressions yields correct results for this table, but we use a predicate to define the bottom edge, as the number of countries can change, and footnotes may be present at the bottom of the table. The predicate `bottom: value(B:/$row) = ""` states that the bottom edge is the first row where columns after `B` are empty.[6] The example also illustrates the ability to skip rows and cells using Boolean expressions. The cells in the data region are highlighted in green.

The `template` section defines the mapping of cell to elements of a statement. The T2WML tool instantiates the template once for every cell defined in the `region` section, binding the variables `$col` and `$row` to the coordinates of the cell being processed. To facilitate understanding of the template instantiation procedure, users can click on a data cell in the table viewer to see how it is mapped. The interface shows the values of `$col` and `$row`, highlights the cell containing the item (subject) of the statement (blue), the cells containing the qualifiers (pink), and shows the resulting statement in an output panel (bottom right of Fig. 2).

Users define the relationships between a value cell and other parts of a statement in the YAML editor using expressions defined above. Fig. 2 illustrates the definition of the subject, predicate, value and qualifiers of a statement, which we summarize below:

Subject: line 11 defines the subject of the statement for a value cell as the item in the same row in column A. For the value in cell F6, the item is Burundi, also shown in the output panel.

Predicate: the predicate of the statement is specified as a constant (`P100024`, defined in our clone of Wikidata). It is also possible to define the properties using the `item` function, a convenient feature to map spreadsheets where different columns contain information about different properties.

Value: the value of a statement is usually defined using the expression `value($col/$row)`, or function that transforms the value. T2WML offers a library of string, data and numeric transformation functions similar to those provided in spreadsheet software.

Qualifiers: the qualifiers of a statement are defined in the `qualifiers` section of the YAML file. Each qualifier consists of a predicate and a value. Lines 15 through 19 illustrate the definition of a time qualifier including specification of the value, calendar, precision and time zone.

References: references are defined in the `references` section of the YAML file (not shown in the figure), and are defined similarly to qualifiers.

[6] The : operator is a shortcut for the Boolean *and* operator.

4 DISCUSSION

T2WML is a new language and system under active development. While our experience with T2WML is limited, the results so far are encouraging.

We evaluated the expressivity of T2WML by creating 19 variants[7] of the homicides table downloaded from dataunodc.un.org. We created the variants from the Database layout (Fig. 1c), moving the qualifiers into header rows to emulate common layouts, including multiple header rows (Fig. 1b). We created stacked table variants (Fig. 1d) as tables of this type are also common in web sites. T2WML can map all variants except one where we combined the female and male homicide values in a single cell, separated by comma.

We used T2WML to create Wikidata statements for 9 World Bank indicators (all countries, all available years), and for the Fragile States Index indicators (https://fragilestatesindex.org). Political scientists are using these statements to build models querying our Wikidata clone.

We also used T2WML to map county-level crime data from the FBI.[8] We had originally written a 400-line Python script to map this data, and were able to replicate the results using a T2WML file of the same complexity as the one shown in Fig. 2.

Finally, we evaluated the usability of T2WML with a second-year undergraduate student who is creating Wikidata statements for public crime records in Los Angeles. The dataset contains over 2 million records and is updated weekly. Each crime record is highly detailed, including information such as the address, type of crime, time of day and location, source, etc. Despite not being an expert in mapping languages or RDF, the student was able to use T2WML to link crime records to Wikidata locations using Wikidata properties.

Our current and future work will focus on four directions. First, T2WML can create statements, but currently cannot create new Wikidata items or properties, or define new labels, aliases and descriptions. We will extend T2WML to support these tasks. Second, we want to integrate T2WML more tightly with Wikidata to support semantic operators to enable users to refer to elements of a table semantically (e.g., the column containing countries). Third, we are investigating machine learning approaches for property recommendation as we found that identifying the correct properties to use is the most difficult part of the mapping process. Finally, we are investigating ideas for publishing T2WML files in Wikidata to crowdsource the creation of mapping files for the significant number of spreadsheet and CSV files that exist on the web.

[7] available at https://github.com/usc-isi-i2/t2wml/tree/master/Datasets
[8] https://bit.ly/2YdBrpn

REFERENCES

[1] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. [n. d.]. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data.

[2] Ivan Ermilov, Sören Auer, and Claus Stadler. 2013. Csv2rdf: User-driven csv to rdf mass conversion framework. In *Proceedings of the ISEM*, Vol. 13. 04–06.

[3] Shubham Gupta, Pedro Szekely, Craig A Knoblock, Aman Goel, Mohsen Taheriyan, and Maria Muslea. 2012. Karma: A system for mapping structured sources into the Semantic Web. In *Extended Semantic Web Conference*. Springer, 430–434.

[4] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* 57, 10 (Sept. 2014), 78–85. https://doi.org/10.1145/2629489