Semi-Automatic Mapping and Extraction of RDF Triples From Wikipedia Tables

Adriana Concha[1], Aidan Hogan[1]

[1]DCC, Universidad de Chile; IMFD; Santiago, Chile

Abstract

Tables within Wikipedia articles contain a substantial volume of rich information. Unfortunately this information is difficult to exploit due to the large number of tables on Wikipedia, as well as the diverse schemas and formats these tables use in order to be visually appealing to users. Consequently, manual extraction of structured information from these tables becomes impractical, and automating the process becomes very complex. To address this, our solution suggests clustering tables with similar headers and employing mapping languages for structured information extraction from these clusters. We compare mapping languages and processors for semantic relation extraction from Wikipedia tables, aiming to enhance integration with knowledge graphs like Wikidata. Our analysis – unique for such a data corpus – identifies Tarql as the most efficient processor, generating 984,260 triples from the top ten largest clusters by the number of tables, with 791,021 being novel in Wikidata. Assessing 500 relations, we achieve an average precision of 84.6%, improving the precision of previous automated methods at 81.5% and 70%. Excluding specific clusters could further improve precision.

Keywords

Web Tables, Wikipedia, Wikidata, Mapping languages, Knowledge graphs, Information extraction

1. Introduction

Wikipedia, a multilingual online free encyclopedia, contains over 6.8 million articles within its English edition[1], which are collaboratively edited by contributors worldwide. These articles contain millions of tables filled with factual data [1]. Despite the huge amount of information contained in Wikipedia, there are several issues that compromise the reliability and accessibility of Wikipedia data. The content in Wikipedia articles is meant for human consumption, so while the content is understandable and visually appealing for the human eye, the lack of structure hinders the automatic extraction of data.

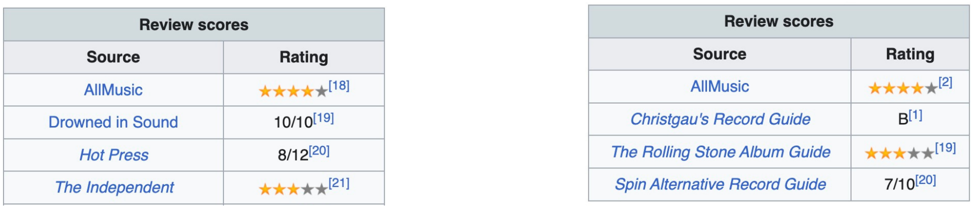
Knowledge graphs like DBpedia [2] and YAGO [3] have been built on Wikipedia data, focusing primarily on extracting semantic triples from Wikipedia's infoboxes. Addressing issues with how Wikipedia manages its data, the Wikidata [4] knowledge graph was proposed to centralize the collection, curation and management of structured data relating to Wikimedia initiatives, and is now used in selected cases to populate infoboxes and other elements of Wikipedia. However, information in such knowledge graphs remains incomplete. There is still a wealth of information in Wikipedia articles – including in the tables of Wikipedia – that is not reflected in such knowledge graphs [1], and cannot be used in a meaningful way other than to be read by humans. This motivates work on extracting semantic triples from such tables [5, 6, 7, 8, 9] that can be used to enrich existing knowledge graphs, and thereafter, to enrich the applications that depend upon them.

Extracting and categorizing semantic triples from Wikipedia tables is a challenging task, as the tables

AMW 2024: 16th Alberto Mendelzon International Workshop on Foundations of Data Management, September 30th–October 4th, 2024, Mexico City, Mexico

adrianaconcha.s@gmail.com (A. Concha); ahogan@dcc.uchile.cl (A. Hogan)
https://aidanhogan.com/ (A. Hogan)
00000-0001-9482-1982 (A. Hogan)

[1] https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia. (Accessed: August 16th, 2024).



Review scores	
Source	Rating
AllMusic	★★★★☆[18]
Drowned in Sound	10/10[19]
Hot Press	8/12[20]
The Independent	★★★☆☆[21]

(a) Table 1

Review scores	
Source	Rating
AllMusic	★★★★☆[2]
Christgau's Record Guide	B[1]
The Rolling Stone Album Guide	★★★☆☆[19]
Spin Alternative Record Guide	7/10[20]

(b) Table 2

Figure 1: Excerpts of tables sharing a cluster (sources: (a) Origin of Symmetry & (b) Suffer (album) on Wikipedia)

have very diverse structures and formats. Previous research has attempted to address this problem using various automated techniques such as machine learning [5], probabilistic graphical models [6], knowledge-based approaches [7, 8], and table clustering [9]. Despite these efforts, the precision of the triples extracted using such automated methods is below the typical precision expected for knowledge graphs like DBpedia, Wikidata and YAGO. Therefore, further improvements are necessary to enhance the accuracy and reliability of these extraction methods in order to increase the potential for successfully incorporating the extracted triples into existing knowledge graphs.

Based on previous research, we explore a different approach towards extracting a large corpus of triples with higher precision from Wikipedia. Specifically, we propose a solution that involves applying user-defined mappings over the tables of Wikipedia in order to extract triples. Given that there are millions of such tables, it is unreasonable to assume that users will define a mapping for each table. Thus we rather propose and explore a method that applies mappings over clusters of Wikipedia tables grouped by similar headers [9]. Such clusters form naturally due to the use of templates for tables in Wikipedia, as well as the editorial practices of copying-and-pasting similar table structures between articles. In our proposal, one mapping can be defined for each cluster, which may contain thousands of tables and result in multitudinous triples being extracted at (we hypothesize) high precision.

A key part of this approach is to choose a suitable mapping language for the Wikipedia setting, which has some idiosyncratic requirements not often considered. In this paper, we compare various RDF mapping language processors, applying them over clusters of Wikipedia tables. We evaluate the processors' performance and the precision and novelty of the triples extracted in Wikidata. Our contributions are twofold: firstly, we compare RDF mapping languages and processors in a real-world setting, and secondly, we propose a novel framework for extracting triples from Wikipedia tables.

Motivating example: Figure 1 displays two Wikipedia tables that share the same schema. These tables were grouped by Luzuriaga et al. [1] into the same cluster of tables, alongside others tables with the same schema. Table 1 provides a sample of post-processed tabular data available for this cluster, where formatting is removed and links in Wikipedia tables are used by Luzuriaga et al. [1] to identify relevant Wikidata entities. In this sample, the entity of the Wikipedia article where the table is embedded is prepended as a protagonist column. For this cluster, we would like to define a single mapping to be applied to each table of the cluster to extract RDF triples following the graph pattern shown in Figure 2. To extract these relations, we need a mapping language and processor capable of managing n-ary relations and being able to process literal values to filter noise within the rating column. Other clusters will exhibit distinct requirements for a comprehensive extraction of triples.

2. Related Work

We discuss previous works relating to Web tables and Mapping languages.

Table 1
Sample from the cluster containing Review scores

ArticleEntity	Review_score_Source	Review_score__Rating
Suffer[Q1537584]	AllMusic[Q31181]	
Suffer[Q1537584]	Spin Alternative Record Guide[Q20875838]	7/10
Origin of Symmetry[Q210910]	AllMusic[Q31181]	
Origin of Symmetry[Q210910]	Drowned in Sound[Q3040076]	10/10

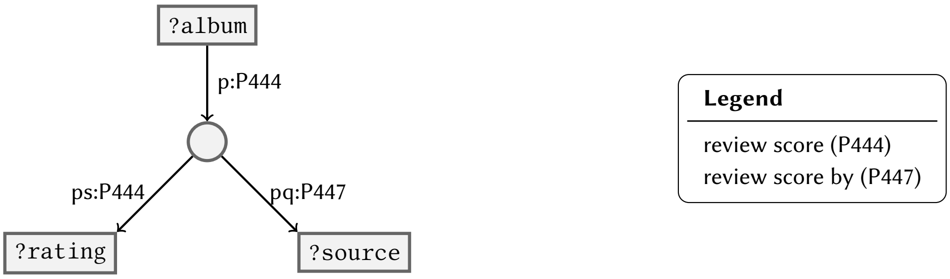


Figure 2: Graph pattern of expected triples for Cluster containing Review scores

2.1. Web tables

Extracting information from Web tables involves two main tasks: table detection and interpretation.

Table detection and categorization In a first task, tables are identified and classified into various groups depending on their taxonomy. Liu et al. (2023) [10] present a novel classification of Web tables based on existing research. They categorize genuine tables, which contain meaningful semantic relationships, across three non-mutually exclusive dimensions: structure, inner relationship, and orientation. The identified types of tables include nested tables, split tables, multivalued tables, entity tables, matrix tables, and horizontal and vertical tables.

Table interpretation The goal of this step is to identify the entities and relationships present within the tables. This involves parsing and normalizing the tables. Next, the entities and attributes within the table cells are identified, and relations between previously identified entities and attributes are extracted. Works typically focus on interpreting horizontal relational tables only (per, e.g., Figure 1). Here we focus on some of the most relevant approaches proposed for Wikipedia tables.

Limaye et al. (2010) [5] propose a method that applies machine learning techniques to annotate Web tables in the YAGO knowledge base. To do this, they annotate the entities contained in each table cell, associate each table column with a type, and extract relationships between pairs of columns. The method proposed by Limaye et al. (2010) for a dataset of Wikipedia tables (excluding infoboxes), reaches an accuracy of 83% for entity annotation, 56% for type annotation and 68% for relation annotation.

Another method that infers types and relationships between pairs of columns is proposed by Mulwad et al. (2013) [6], through a probabilistic graphical model. Their approach outperformed the method proposed by Limaye et al. (2010) in terms of accuracy, achieving a relation annotation F-score of 97% compared to the 68% achieved by Limaye et al. However, it should be noted that the experiment was conducted on a limited dataset of only 36 non-infobox tables extracted from Wikipedia.

Muñoz et al. (2013) [7] propose extracting relations that exist in DBpedia between entities in different columns of the same row and then extending that relation to other rows of the table. With this method, 24.4 million raw triples were extracted from Wikipedia's tables with an estimated precision of 52%. Muñoz et al. (2014) [8] extend the previous method, testing machine learning methods for classifying correct/incorrect triples, extracting 7.9 million novel RDF triples over one million Wikipedia tables with an estimated precision of 81.5% and a F-score of 79.4%.

Luzuriaga et al. (2023) [1] extend the work of Muñoz et al. (2014) by first clustering tables according to their content and structure, with the aim of increasing the quantity and precision of the relations extracted. This method extracted 7.5 million novel triples for Wikidata over a more up-to-date collection of 3.6 million Wikipedia tables, reaching a precision of 70%.

2.2. Mapping languages to RDF

Iglesias-Molina et al. (2022) [11] categorize RDF mapping languages into three groups based on syntax: RDF-based, SPARQL-based, and others. Our focus will be on the first two categories.

RDF-based mapping languages These mapping languages are defined as ontologies. One example is R2RML [12]: a mapping language for defining custom mappings from relational data to RDF. The mappings generated with this language enable viewing relational data as RDF, with a structure and lexicon customized by the author of the mapping. Another key example is RML (RDF Mapping Language) [13], inspired by R2RML, which is a language used to define mappings between non-RDF sources and RDF. RML is more general than R2RML, and allows for defining mappings from several semi-structured forms (JSON, XML, CSV) to RDF, rather than only from relational databases.

Several tools implement the RML specification. The RML Implementation report[2] evaluates some of these tools, testing their coverage of RML's features. The tools tested are RMLMapper [14], CARML[3], RocketRML [15], SDM-RDFizer [16], RMLStreamer [17], Chimera [18], and Morph-KGC [19].

SPARQL-based mapping languages Several mapping languages are based on the SPARQL query language whose CONSTRUCT feature can convert tables (typically of solutions) into RDF graphs. Some of them are Tarql[4], SPARQL-Generate [20], SPARQL Anything [21], XSPARQL [22] and SMS2[5].

2.3. Novelty

Previous studies have primarily focused on developing automated techniques for interpreting web tables, particularly infoboxes from Wikipedia or a limited number of Wikipedia tables that are not infoboxes. However, these methods face a trade-off between the precision and recall of the relationships extracted. Manual methods, while accurate, require significant human effort to map tables individually. This process can be tedious and time-consuming, especially when dealing with a large number of tables. Our goal is to extract RDF triples from Wikipedia tables with high precision at large scale. We propose a novel method using hand-crafted RDF mappings applied to clusters of tables instead of individual tables. We evaluate this method by comparing various mapping languages to RDF.

3. Data corpus

The data corpus, prepared by Luzuriaga [9] in 2019, includes 3,631,228 (non-infobox) Wikitables from 997,842 English Wikipedia articles. Among these articles, the one with the most tables contains 676,

[2] https://rml.io/implementation-report/

[3] https://github.com/carml/carml

[4] https://tarql.github.io/

[5] https://docs.stardog.com/virtual-graphs/mapping-data-sources#sms2-stardog-mapping-syntax-2

Table 2
Top 10 clusters by № of tables

Cluster ID	№ of Tables	№ of Cols.	№ of Rows	№ of Rows with multiple entities	Columns
1	81,277	6	1,202,635	526,939	Party, Party, Candidate, Votes, %, ±
2	57,878	5	650,861	283,968	Party, Party, Candidate, Votes, %
3	57,427	2	187,707	106,544	Review scores**Source, Review scores**Rating
4	56,332	3	450,119	44,906	No.,Title, Length
5	51,229	4	496,619	495,033	No., spancol, Position, Player
6	25,449	6	50,898	340	spancol, 1, 2, 3, 4, Total
7	23,472	4	459,839	73,910	Notes, Year, Title, Role
8	20,772	3	65,300	65,226	Ship, Country, Description
9	17,204	4	138,504	36,345	No., Title, Writer(s), Length
10	15,596	2	128,267	120,208	Position, Player

while 51.6% have only a single table. A total of 3,514,373 tables (35,428,465 rows) have an associated Wikidata entity for the article, which is added as a protagonist column that often has pertinent relations to other columns [23] (per ArticleEntity in Table 1).

The dataset consists of 1,169,682 clusters of tables that share the same schema. These schemas are defined by a subset of attributes, ensuring that all tables within a cluster adhere to this common schema [1]. While most clusters (82.7%) contain only one table, some extensive clusters contain a considerable number of tables. Table 2 lists the top 10 clusters by their number of tables, including data about the columns, rows, and the number of rows with multiple entities across the columns (a key target for extracting triples). This analysis reveals the presence of substantial clusters from which a large amount of data can be extracted. For example, the largest cluster contains 81,277 tables and 1,202,635 rows, with 536,939 rows exhibiting multi-column entities. Moreover, the columns that feature literal values can serve as objects in newly extracted relationships. As such, with a single mapping, it may be feasible to extract more than half a million triples from this cluster.

4. Features of mapping languages

We selected mapping languages and processors based on specific criteria. The chosen language must accept CSV input and have an open-source processor (Table 3). For RML processors, we prioritized those with top results in the RML Implementation report, focusing on tests related to blank node generation for reliable extraction of *n*-ary relations (Table 4). We excluded the use of custom functions, such as those in RML + FnO. Despite RMLStreamer's success in tests, it had issues with CSV input length limits and failed to use the bulk option for writing triples from a single input record.

Analyzing these top 10 clusters, we identified specific requisites for extracting information from horizontal relational tables, multivalued tables, and matrix tables. The features are defined as follows, while Table 5 summarizes the ability of the selected tools to fulfill these requirements. We publish mappings testing these features on GitHub[7] for the top 10 largest clusters of Wikipedia tables, with Wikidata as a target. To satisfy the requirements for fulfilling these features, we primarily use SPARQL functions in SPARQL-based mapping languages and SQL in RML Views [24], implemented by Morph-KGC. We also used built-in functions provided by RMLMapper and Morph-KGC.

[6] The implementation is currently unavailable

[7] https://github.com/AdrianaConcha/Wikitables-RDF-mappings

Table 3
Mapping Languages selection criteria

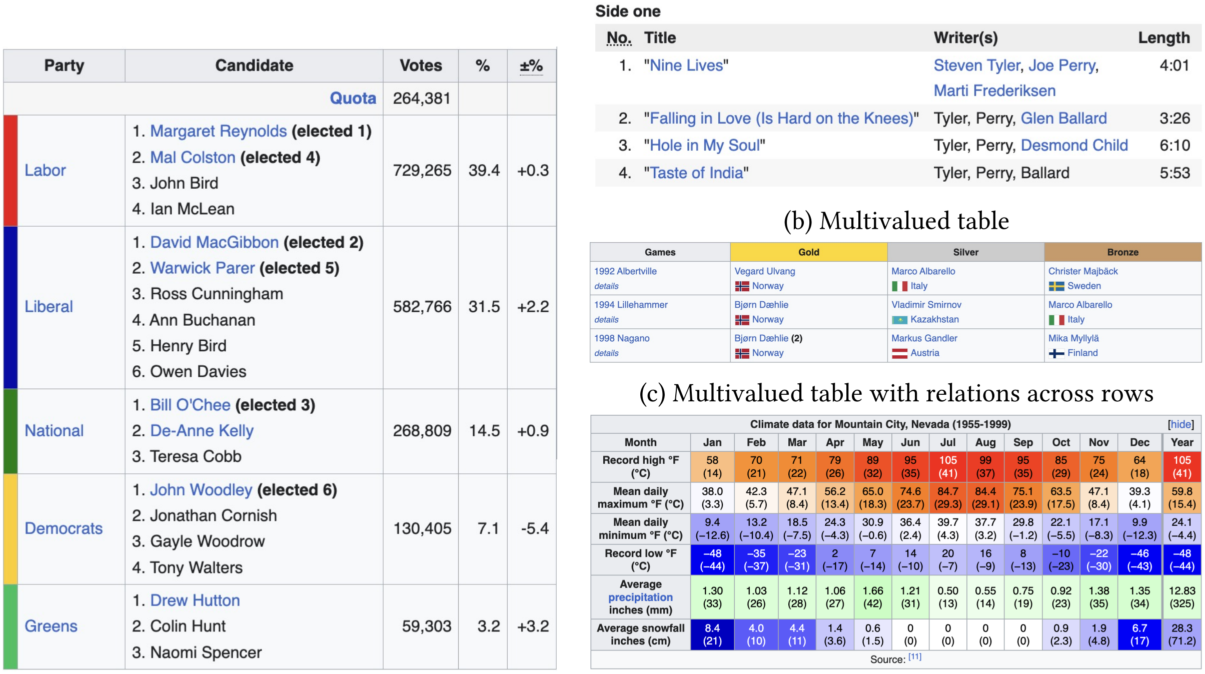
Mapping Language	CSV	Open Source	Tool
R2RML	✗	✓	✓
RML	✓	✓	✓
FunUL	✓	✓	✗[6]
D2RML	✓	✗	✓
Tarql	✓	✓	✓
SPARQL-Generate	✓	✓	✓
SPARQL Anything	✓	✓	✓
XSPARQL	✗	✓	✓
SMS2	✓	✗	✓

Table 4
RML processors selection criteria

RML engine	Blank nodes test passed	CSV no length limit
RMLMapper	✓	✓
CARML	✓	✓
RocketRML	✗	✓
SDM-RDFizer	✓	✓
RMLStreamer	✓	✗
Chimera	✓	✓
Morph-KGC	✓	✓

- **F1: Extraction of triples between columns:** A basic requirement is the ability to extract a triple with a subject in one column and the object entity in another column within the same row in horizontal tables using a single predicate type. For example, we wish to extract the relation Drew Hutton[Q5307201] member of political party[P102] Greens[Q781486] from Figure 3a.
- **F2: Extraction of relations considering all entities within a single cell:** In multivalued tables, we need to extract relations for all entities within a cell. This requires the tool to split the string in the multivalued cell. For example, in Figure 3a, first row, we want to extract the triples Margaret Reynolds[Q6759833] member of political party[P102] Labor[Q216082] and Mal Colston[Q15506241] member of political party[P102] Labor[Q216082].
- **F3: Extraction of relations considering the n^{th} entity within a cell:** In multivalued tables, extracting relations for the n^{th} entity in a cell requires splitting the cell content and selecting the appropriate index or processing the output accordingly. For instance, in Figure 3c, we wish to extract Vegard Ulvang[Q370499] participant in[P1344] 1992 Albertville[Q1042417] from the first entity in the "Gold" column of the first row, ignoring the second entity (that indicates a country).
- **F4: Extraction of relations between entities within a single cell:** Similar to F3, this requires splitting the cell content and processing the output to extract relations, for instance, extracting the triple Vegard Ulvang[Q370499] country of citizenship[P27] Norway[Q20] from Figure 3c.
- **F5: Extraction of *n*-ary relations without defining a template:**[8] Due to potentially missing values in tables, ensuring a distinct key for each *n*-ary relation isn't feasible. Hence, we evaluate the processor's capability to generate blank node identifiers without needing a predefined template.
- **F6: Extraction of relations across different rows:** This requirement involves extracting a triple with a subject in one row and the object entity in the following row within the same column. To achieve this, the tool needs to support join operations, provide row identifiers to sequence rows, and allow arithmetic operations. For example, in Figure 3c, we aim to extract the triple 1992 Albertville[Q1042417] followed by[P156] 1994 Lillehammer[Q602473].
- **F7: Extraction of relations in matrix tables:** This requirement evaluates the ability of the tools to extract *n*-ary relations for an entity in a cell, considering the headers of its respective column and row, for example, extracting relations from Figure 3d.
- **F8: Handling of literal values within the mapping:** We often need to preprocess and manage literal values (using built-in functions or other available functions), for example, to convert the MM:SS values in the "Length" column of Figure 3b to seconds.

[8] We acknowledge that the RML specification stipulates that a term map (constant, column/reference or a template) must be provided while generating blank nodes, but this constraint doesn't align with our use case.



Party	Candidate	Votes	%	±%
	Quota	264,381		
Labor	1. Margaret Reynolds **(elected 1)** 2. Mal Colston **(elected 4)** 3. John Bird 4. Ian McLean	729,265	39.4	+0.3
Liberal	1. David MacGibbon **(elected 2)** 2. Warwick Parer **(elected 5)** 3. Ross Cunningham 4. Ann Buchanan 5. Henry Bird 6. Owen Davies	582,766	31.5	+2.2
National	1. Bill O'Chee **(elected 3)** 2. De-Anne Kelly 3. Teresa Cobb	268,809	14.5	+0.9
Democrats	1. John Woodley **(elected 6)** 2. Jonathan Cornish 3. Gayle Woodrow 4. Tony Walters	130,405	7.1	-5.4
Greens	1. Drew Hutton 2. Colin Hunt 3. Naomi Spencer	59,303	3.2	+3.2

(a) Horizontal relational multivalued table

Side one

No.	Title	Writer(s)	Length
1.	"Nine Lives"	Steven Tyler, Joe Perry, Marti Frederiksen	4:01
2.	"Falling in Love (Is Hard on the Knees)"	Tyler, Perry, Glen Ballard	3:26
3.	"Hole in My Soul"	Tyler, Perry, Desmond Child	6:10
4.	"Taste of India"	Tyler, Perry, Ballard	5:53

(b) Multivalued table

Games	Gold	Silver	Bronze
1992 Albertville *details*	Vegard Ulvang Norway	Marco Albarello Italy	Christer Majbäck Sweden
1994 Lillehammer *details*	Bjørn Dæhlie Norway	Vladimir Smirnov Kazakhstan	Marco Albarello Italy
1998 Nagano *details*	Bjørn Dæhlie **(2)** Norway	Markus Gandler Austria	Mika Myllylä Finland

(c) Multivalued table with relations across rows

Climate data for Mountain City, Nevada (1955-1999) [hide]

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Year
Record high °F (°C)	58 (14)	70 (21)	71 (22)	79 (26)	89 (32)	95 (35)	105 (41)	99 (37)	95 (35)	85 (29)	75 (24)	64 (18)	105 (41)
Mean daily maximum °F (°C)	38.0 (3.3)	42.3 (5.7)	47.1 (8.4)	56.2 (13.4)	65.0 (18.3)	74.6 (23.7)	84.7 (29.3)	84.4 (29.1)	75.1 (23.9)	63.5 (17.5)	47.1 (8.4)	39.3 (4.1)	59.8 (15.4)
Mean daily minimum °F (°C)	9.4 (−12.6)	13.2 (−10.4)	18.5 (−7.5)	24.3 (−4.3)	30.9 (−0.6)	36.4 (2.4)	39.7 (4.3)	37.7 (3.2)	29.8 (−1.2)	22.1 (−5.5)	17.1 (−8.3)	9.9 (−12.3)	24.1 (−4.4)
Record low °F (°C)	−48 (−44)	−35 (−37)	−23 (−31)	2 (−17)	7 (−14)	14 (−10)	20 (−7)	16 (−9)	8 (−13)	−10 (−23)	−22 (−30)	−46 (−43)	−48 (−44)
Average precipitation inches (mm)	1.30 (33)	1.03 (26)	1.12 (28)	1.06 (27)	1.66 (42)	1.21 (31)	0.50 (13)	0.55 (14)	0.75 (19)	0.92 (23)	1.38 (35)	1.35 (34)	12.83 (325)
Average snowfall inches (cm)	8.4 (21)	4.0 (10)	4.4 (11)	1.4 (3.6)	0.6 (1.5)	0 (0)	0 (0)	0 (0)	0 (0)	0.9 (2.3)	1.9 (4.8)	6.7 (17)	28.3 (71.2)

Source: [11]

(d) Matrix table

Figure 3: Table examples (sources: (a) 1993 Australian Senate election, (b) Nine Lives (Aerosmith album), (c) List of Olympic medalists in cross-country skiing, (d) Mountain City, Nevada on Wikpedia)

Table 5
Features supported by different languages and processor tools

Language	Tool	F1	F2	F3	F4	F5	F6	F7	F8
Tarql	Tarql	✓	✓	✓	✓	✓	✗	✓	✓
SPARQL-Generate	SPARQL-Generate	✓	✓	✓	✓	✓	✗	✓	✓
SPARQL Anything	SPARQL Anything	✓	✓	✓	✓	✓	✗	✓	✓
RML (no FNO)	CARML	✓	✗	✗	✗	✗	✗	✗	✗
RML (no FNO)	RMLMapper	✓	* [9]	✗	✗	✓	✗	✗	* [10]
RML (no FNO)	SDM-RDFizer	✓	✗	✗	✗	✗	✗	✗	✗
RML (no FNO)	Chimera	✓	✗	✗	✗	✓	✗	✗	✗
RML (no FNO)	Morph-KGC	✓	✓	✗	✗	✗	✗	✗	* [10]
RML (no FNO)	Morph-KGC + RML Views	✓	✓	✓	✓	✗	✓	✓	✓

5. Comparison of mapping languages and processors

Using the mappings we previously defined for the top 10 clusters, we now compare the performance and output of different languages and processors.

Figure 4 compares execution times for the top ten largest clusters and each tested processor. The timeout (TMO) of 3600 seconds is set lower in the figure for visualization purposes. Tarql and CARML had the fastest execution times in most experiments. However, CARML sometimes extracted fewer triples due to the need for a blank node template, an issue also seen in SDM-RDFizer and Morph-KGC

[9] The default function grel:string_split appears to not be working in RMLMapper v6.2.1

[10] This tool provides some built-in functions for handling strings, but they were either not used or not functioning correctly during the experiments.

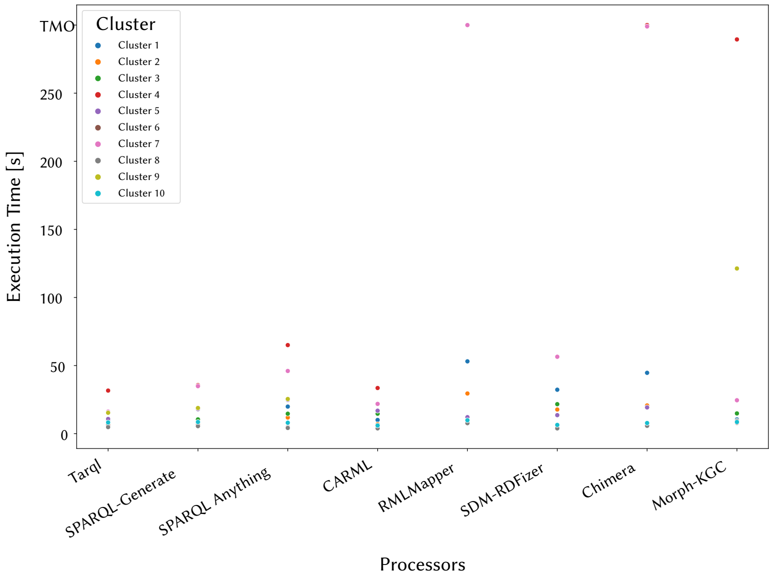
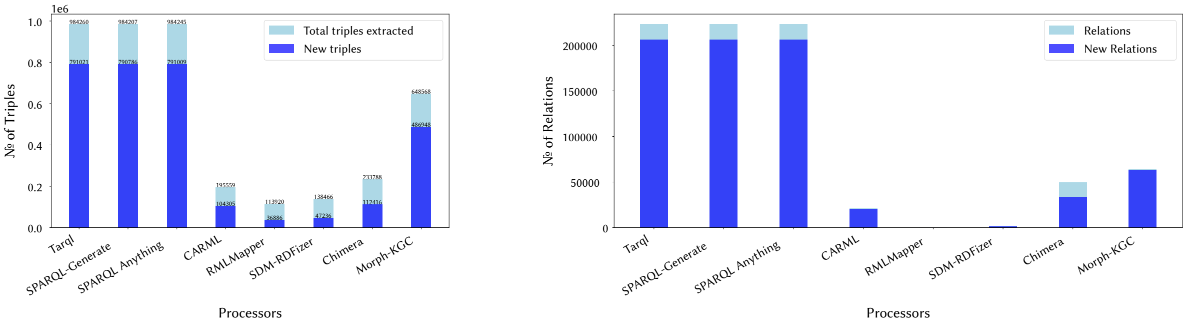


Figure 4: Execution Time for each Cluster and Processor



(a) New and total triples extracted

(b) New and total relations extracted

Figure 5: Comparison of new triples and new relations extracted by each processor over the top 10 clusters

mappings, as shown in Figure 5a. Additionally, SDM-RDFizer and Morph-KGC encountered illegal characters in blank node templates, unsupported by RDFLib, necessitating data preprocessing.

Figure 5a compares the number of triples extracted per processor for the top 10 clusters. Each stacked bar also shows new triples not present in Wikidata. SPARQL-based mapping languages extracted the most triples, with Tarql extracting 984,260 triples, including 791,021 novel ones.

Figure 5b compares the number of relations extracted per processor for the top 10 clusters. The number of relations is fewer than the number of triples since a relation may contain multiple triples.

We also evaluated the precision of novel relations extracted from the top 10 clusters. These novel relations are those not previously present in Wikidata. Within each of these clusters, we randomly sampled 50 relations from the output of Tarql, the mapping language with the most extracted triples, to manually assess their precision, resulting in a total of 500 relations evaluated. Precision is calculated as the ratio of correct extracted relations to the total number of extracted relations. Table 6 presents the results obtained. We classify relations as either correct or incorrect. Relations that require additional

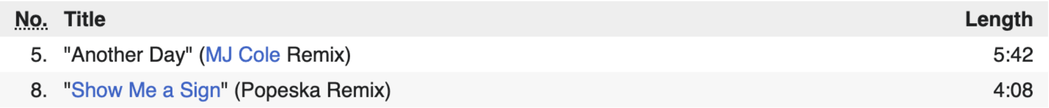
Table 6
Evaluation of extracted triples from the Top 10 largest clusters

Cluster	Correct	Incorrect	Precision [%]	Reasons for Incorrectness	№ of incorrect relations per reason
1	50	0	100	–	
2	46	4	92	Incorrect Wikipedia link	4
3	50	0	100	–	
4	25	25	50	Subject or object doesn't have an entity Incorrect assumption in mapping Incorrect Wikipedia link	21 2 2
5	43	7	86	Subject or object doesn't have an entity Incorrect Wikipedia link	4 3
6	42	8	84	Incorrect assumption in mapping Incorrect Wikipedia link	7 1
7	50	0	100	–	
8	43	7	86	Columns with diverse entity types Incorrect Wikipedia link	6 1
9	40	10	80	Subject or object doesn't have an entity Incorrect assumption in mapping	8 2
10	34	16	68	Incorrect Wikipedia link Subject or object doesn't have an entity	9 7

qualifiers to fully convey the information are still deemed correct due to the open world assumption in Wikidata, where missing information is treated as unknown rather than false. Consequently, such relations are considered valid, with the potential for these qualifiers to be added in future.

Some of the reasons why relations were considered as incorrect are as follows:

- **Incorrect Wikipedia link:** The link in the Wikipedia table (from which the Wikidata entity was extracted by Luzuriaga [9]) is incorrect. For example, the triple Gerardo Flores[Q5550262] position played on team[P413] defender[Q336286] extracted from Cluster 5 is correct for the subject Gerardo Flores[Q5550269] the footballer, but the entity extracted was Gerardo Flores[Q5550262] the murderer.
- **Subject or object doesn't have an entity:** The subject or object of the triple doesn't have a Wikidata entity, but there is a different entity in the same cell. For example, the song "Another Day" from Cluster 4 doesn't have a Wikidata entity. Therefore, the entity extracted in the "Title" column in Figure 6 for that cell was M. J. Cole[Q708129]. When we executed the mapping for this cluster we obtained the incorrect triple M. J. Cole[Q708129] form of creative work[P7937] song[Q7366].

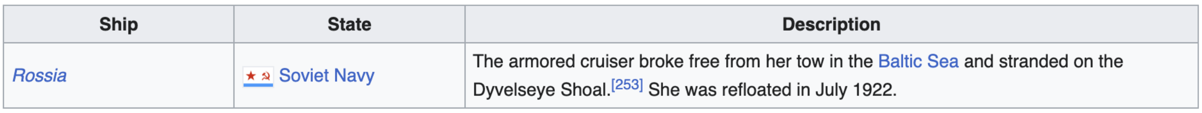


No.	Title	Length
5.	"Another Day" (MJ Cole Remix)	5:42
8.	"Show Me a Sign" (Popeska Remix)	4:08

Figure 6: Subject doesn't have an entity (abridged from Evolution Theory (Modestep album) on Wikipedia)

- **Incorrect assumption in mapping:** When defining the mappings, we make some assumptions about the articles from which the tables were extracted. While in most cases these assumptions were accurate, in some cases they were not. For instance, we assume that articles in Cluster 9 refer to albums, but for a table from the "The Time of the Oath (song)" article, we obtained the incorrect triple The Time of the Oath[Q760753] instance of[P31] album[Q482994].
- **Columns with diverse entity types:** A column may contain subtly different types of entities. For example, in Cluster 8, some tables indicate a navy instead of a location in the "State" column.

For example, from the table in Figure 7, we extracted the incorrect triple Rossia[Q690108] country of registry[P8047] Soviet Navy[Q796754]. We also included incorrect facts in this category.



Ship	State	Description
Rossia	Soviet Navy	The armored cruiser broke free from her tow in the Baltic Sea and stranded on the Dyvelseye Shoal.[253] She was refloated in July 1922.

Figure 7: Columns with diverse entity types (abridged from List of shipwrecks in 1922 on Wikipedia)

For clusters 1, 3, and 7, all the sampled relations were correct, obtaining a precision of 100%. The average precision of the sampled relations of the top ten largest clusters is 84.6%.

6. Conclusions

In this work, we propose a novel semi-automatic method for extracting triples from Wikipedia tables at large scale and with high precision. Experimental evaluations involved applying mappings to table clusters, measuring triple and relation extraction, processor performance, and the novelty of extracted triples not present in Wikidata. Our study identified Tarql as the most effective processor, generating 984,260 triples with 791,021 novel to Wikidata. Sampling 50 random relations extracted by Tarql from each of the top ten largest clusters yielded an average precision of 84.6%, surpassing previous methods which achieved 81.5% [8] and 70% [9] precision. However, as the imprecision is primarily associated with specific clusters, excluding these clusters could significantly improve the precision achieved.

We presented a comparative analysis of RDF mapping languages and processors to determine their suitability for extracting RDF triples from Wikipedia tables, building on previous work that merged tables with identical headers. Our analysis of the Wikipedia table corpus provided insights into individual tables and clusters, revealing substantial extractable information. The largest cluster, comprising 81,277 tables with 1,202,635 rows, included 536,939 rows suitable for relation extraction due to multi-column entity relationships. We emphasized the importance of selecting appropriate languages and processors for this data corpus. Through examples with various table schemas – horizontal relational, multivalued, and matrix tables – we assessed the expressiveness and performance of SPARQL- and RML-based mapping languages. SPARQL-based languages and the use of SQL in RML Views showed significant advantages in processing, filtering, and handling complex schemas.

These findings support our hypothesis that applying RDF mapping languages over clusters of tables can extract significant volumes of high-precision novel triples for knowledge graphs like Wikidata.

While this work has provided valuable insights into the extraction of RDF triples from Wikipedia tables, there are some limitations to mention. First, our study used a 2019 dataset by Luzuriaga [9]; updating the corpus could enhance the relevance and accuracy of extracted relations. Moreover, improving the granularity of extracted information, such as entity positions within cells and associating entities with table titles, could further enrich the data. Future work could focus on developing a systematic approach for relation extraction using our findings, empowering expert users to apply mappings across clusters with minimal manual effort. Integrating the novel relations into Wikidata and post-processing the triples to verify entity types could further improve precision.

Another potential direction involves leveraging large language models (LLMs) for relation extraction from Wikipedia tables. Progress has been made in web table interpretation through instruction tuning of LLMs, with a framework like TURL [25] – focused on relational web tables from Wikipedia –showing promise. However, more research is needed to manage varying table schemas and address challenges such as manipulating table data (e.g., handling literal values) and fine-tuning LLMs for this specific corpus of Wikipedia tables [26]. Furthermore, employing LLMs to assist users in creating mappings (per the approach used in this paper) is also a promising avenue for future exploration [27].

Acknowledgments

This work was supported by Fondecyt No. 1221926 and by ANID – Millennium Science Initiative Program – Code ICN17_002.

References

[1] J. Luzuriaga, E. Muñoz, H. Rosales-Méndez, A. Hogan, Merging Web Tables for Relation Extraction With Knowledge Graphs, IEEE Trans. Knowl. Data Eng. 35 (2023) 1803–1816. URL: https://doi.org/10.1109/TKDE.2021.3101479. doi:10.1109/TKDE.2021.3101479.

[2] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, C. Bizer, DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia, Semantic Web 6 (2015) 167–195. URL: https://doi.org/10.3233/SW-140134. doi:10.3233/SW-140134.

[3] J. Hoffart, F. M. Suchanek, K. Berberich, G. Weikum, YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia, Artif. Intell. 194 (2013) 28–61. URL: https://doi.org/10.1016/j.artint.2012.06.001. doi:10.1016/J.ARTINT.2012.06.001.

[4] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Commun. ACM 57 (2014) 78–85. doi:10.1145/2629489.

[5] G. Limaye, S. Sarawagi, S. Chakrabarti, Annotating and Searching Web Tables Using Entities, Types and Relationships, PVLDB 3 (2010) 1338–1347. doi:10.14778/1920841.1921005.

[6] V. Mulwad, T. Finin, A. Joshi, Semantic Message Passing for Generating Linked Data from Tables, in: International Semantic Web Conference (ISWC), Springer, 2013, pp. 363–378. DOI:10.1007/978-3-642-41335-3_23.

[7] E. Muñoz, A. Hogan, A. Mileo, Triplifying Wikipedia's Tables, in: First International Conference on Linked Data for Information Extraction - Volume 1057, LD4IE'13, pages 26–37, Aachen, Germany, Germany, 2013.

[8] E. Muñoz, A. Hogan, A. Mileo, Using linked data to mine RDF from Wikipedia's tables, in: Web Search and Web Data Mining (WSDM), ACM, 2014, pp. 533–542. doi:10.1145/2556195.2556266, dOI:10.1145/2556195.2556266.

[9] J. Luzuriaga, Merging HTML Tables for Extracting Relations, Tesis de Magíster, Universidad de Chile, Santiago, Chile, 2019.

[10] J. Liu, Y. Chabot, R. Troncy, V.-P. Huynh, T. Labbé, P. Monnin, From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods, Journal of Web Semantics 76 (2023) 100761. URL: https://www.sciencedirect.com/science/article/pii/S1570826822000452. doi:https://doi.org/10.1016/j.websem.2022.100761.

[11] A. Iglesias-Molina, A. Cimmino Arriaga, E. Ruckhaus, D. Chaves-Fraga, R. García Castro, O. Corcho, An Ontological Approach for Representing Declarative Mapping Languages, Semantic Web (2022). doi:10.3233/SW-223224.

[12] S. Das, S. Sundara, R. Cyganiak, R2RML: RDB to RDF Mapping Language, https://www.w3.org/TR/r2rml/, 2012.

[13] A. Dimou, M. Vander Sande, B. De Meester, P. Heyvaert, T. Delva, RDF Mapping Language (RML), https://rml.io/specs/rml/, 2022.

[14] A. Dimou, T. D. Nies, R. Verborgh, E. Mannens, R. V. de Walle, Automated Metadata Generation for Linked Data Generation and Publishing Workflows, in: S. Auer, T. Berners-Lee, C. Bizer, T. Heath (Eds.), Proceedings of the Workshop on Linked Data on the Web, LDOW 2016, co-located with 25th International World Wide Web Conference (WWW 2016), volume 1593 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2016. URL: https://ceur-ws.org/Vol-1593/article-04.pdf.

[15] U. Simsek, E. Kärle, D. Fensel, RocketRML - A NodeJS Implementation of a Use Case Specific RML Mapper, in: D. Chaves-Fraga, P. Heyvaert, F. Priyatna, J. F. Sequeda, A. Dimou, H. Jabeen, D. Graux, G. Sejdiu, M. Saleem, J. Lehmann (Eds.), Joint Proceedings of the 1st International Workshop on Knowledge Graph Building and 1st International Workshop on Large Scale RDF Analytics co-located with 16th Extended Semantic Web Conference (ESWC 2019), Portorož, Slovenia, June 3, 2019, volume 2489 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 46–53. URL: https://ceur-ws.org/Vol-2489/paper5.pdf.

[16] E. Iglesias, S. Jozashoori, D. Chaves-Fraga, D. Collarana, M. Vidal, SDM-RDFizer: An RML Interpreter for the Efficient Creation of RDF Knowledge Graphs, in: M. d'Aquin, S. Dietze, C. Hauff, E. Curry, P. Cudré-Mauroux (Eds.), CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, ACM, 2020, pp. 3039–3046. URL: https://doi.org/10.1145/3340531.3412881. doi:10.1145/3340531.3412881.

[17] S. M. Oo, G. Haesendonck, B. D. Meester, A. Dimou, RMLStreamer-SISO: An RDF Stream Generator from Streaming Heterogeneous Data, in: U. Sattler, A. Hogan, C. M. Keet, V. Presutti, J. P. A. Almeida, H. Takeda, P. Monnin, G. Pirrò, C. d'Amato (Eds.), The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings, volume 13489 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 697–713. URL: https://doi.org/10.1007/978-3-031-19433-7_40. doi:10.1007/978-3-031-19433-7_40.

[18] M. Belcao, E. Falzone, E. Bionda, E. D. Valle, Chimera: A Bridge Between Big Data Analytics and Semantic Technologies, in: A. Hotho, E. Blomqvist, S. Dietze, A. Fokoue, Y. Ding, P. M. Barnaghi, A. Haller, M. Dragoni, H. Alani (Eds.), The Semantic Web - ISWC 2021 - 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24-28, 2021, Proceedings, volume 12922 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 463–479. URL: https://doi.org/10.1007/978-3-030-88361-4_27. doi:10.1007/978-3-030-88361-4_27.

[19] J. Arenas-Guerrero, D. Chaves-Fraga, J. Toledo, M. S. Pérez, O. Corcho, Morph-KGC: Scalable knowledge graph materialization with mapping partitions, Semantic Web (2022). doi:10.3233/SW-223135.

[20] M. Lefrançois, A. Zimmermann, N. Bakerally, A SPARQL extension for generating RDF from heterogeneous formats, in: E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, O. Hartig (Eds.), The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I, volume 10249 of *Lecture Notes in Computer Science*, 2017, pp. 35–50. URL: https://doi.org/10.1007/978-3-319-58068-5_3. doi:10.1007/978-3-319-58068-5_3.

[21] L. Asprino, E. Daga, A. Gangemi, P. Mulholland, Knowledge Graph Construction with a Façade: A Unified Method to Access Heterogeneous Data Sources on the Web, ACM Trans. Internet Techn. 23 (2023) 6:1–6:31. URL: https://doi.org/10.1145/3555312. doi:10.1145/3555312.

[22] S. Bischof, S. Decker, T. Krennwallner, N. Lopes, A. Polleres, Mapping between RDF and XML with XSPARQL, J. Data Semant. 1 (2012) 147–185. URL: https://doi.org/10.1007/s13740-012-0008-7. doi:10.1007/S13740-012-0008-7.

[23] E. Crestan, P. Pantel, Web-scale table census and classification, in: I. King, W. Nejdl, H. Li (Eds.), Web Search and Web Data Mining (WSDM), ACM, 2011, pp. 545–554. DOI:10.1145/1935826.1935904.

[24] J. Arenas-Guerrero, A. Alobaid, M. Navas-Loro, M. Pérez, O. Corcho, Boosting Knowledge Graph Generation from Tabular Data with RML Views, in: Proceedings of the 20th Extended Semantic Web Conference, volume 13870, Springer Nature Switzerland, 2023, pp. 484–501. URL: https://link.

springer.com/chapter/10.1007/978-3-031-33455-9%5f29. doi:10.1007/978-3-031-33455-9_29, ontology Engineering Group OEG.

[25] X. Deng, H. Sun, A. Lees, Y. Wu, C. Yu, TURL: Table Understanding through Representation Learning, CoRR abs/2006.14806 (2020). URL: https://arxiv.org/abs/2006.14806. arXiv:2006.14806.

[26] W. Lu, J. Zhang, J. Zhang, Y. Chen, Large Language Model for Table Processing: A Survey, CoRR abs/2402.05121 (2024). URL: https://doi.org/10.48550/arXiv.2402.05121. doi:10.48550/ARXIV.2402.05121. arXiv:2402.05121.

[27] M. Hofer, J. Frey, E. Rahm, Towards self-configuring Knowledge Graph Construction Pipelines using LLMs - A Case Study with RML, in: D. Chaves-Fraga, A. Dimou, A. Iglesias-Molina, U. Serles, D. V. Assche (Eds.), Proceedings of the 5th International Workshop on Knowledge Graph Construction co-located with 21th Extended Semantic Web Conference (ESWC 2024), Hersonissos, Greece, May 27, 2024, volume 3718 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3718/paper6.pdf.