TSV2RDF: Generating RDF Data Model from TSV File Format Using Semantic Web Technologies

Mammadov Hasan[1], Yan Li[1], Muhammad Waqas Ahmad[2]
[1]College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[2]College of Electrical and Mechanical Engineering
National University of Sciences and Technology, Islamabad, Pakistan

Journal of Digital Information Management

ABSTRACT: *The Semantic Web empowers machines to better understand the information by associating precise meanings with the data. Resource Description Framework (RDF) lies among basic building blocks for the Semantic Web, to give formal definitions to the data. However humongous amount of majority governments and official data exist in various legacy file formats, generally in the tabular form such as tab-separated values (TSV) or relational databases. Nevertheless, there existed no built-in or standardized implementation to convert TSV to RDF using Semantic Web technologies. This paper focused on deliberation to define and implement a novel mechanism for data conversion. The proposed technique involved in conversion of TSV data to RDF format based on RDF Mapping Language (RML)with direct mapping technique. RML is the RDF mapping language, which is meant to define mapping functions from heterogeneous file formats to RDF data structure. In this paper, an RML and direct mapping based RDF generation software TSV2RDF has been created. TSV2RDF has been evaluated at various benchmark TSV datasets. The proposed conversion method has produced comparative results with increased accuracy and reduced processing time. The performance of developed software exhibits significant performance compared with other datasets. Significant usability at TSV to RDF transformation is suggested by smart adoption of the TSV2RDF.*

Subject Categories and Descriptors: [H.2.4 Systems Relational databases]; [H.2.3 Languages]; [H.3.5 Online Information Services]: Data sharing

General Terms: Semantic Web, Data Sharing, RDF

Keywords: RDF, TSV, RML, Relational to RDF Mapping Language (R2RML), Semantic Web

Received: 4 June 2020, Revised 3 October 2020, Accepted 24 November 2020

Review Metrics: Review Scale: 0-6, Review Score: 4.942, Inter-reviewer consistency: 82.5%

DOI: 10.6025/jdim/2021/19/1/10-26

1. Introduction

Semantic Web is the add-on of standard World Wide Web. It is being regularized by the esteemed World Wide Web Consortium (W3C). Ultimate objective of the Semantic Web is to enable computers and machines to understand the meanings of data over internet [1]. It is a journey from documents' web to data's web, through W3C's vision for web of linked data. Semantic Web comes up with a ubiquitous framework that enables data sharing among heterogeneous application, diverse users and different platforms. Hence, it is considered as an integrator for various contents, information systems and applications [2].

Resource Description Framework (RDF) is set of W3C specifications, primarily designed as a tool for metadata definitions. RDF is the basic enabling technology for encoding the semantics with data. It is an architecture that provides the mechanism for encoding, interchange and reuse of metadata in a structured way [3]. The RDF data modeling is conceptually similar to the conventional modeling approaches (like entity relationship diagram and UML class diagrams). The core idea is to express resources/web resources as statements. These statements

make expressions comprising of subject, predicate and object, which are then referred as triples [4]. The resources/ web resources are denoted by subjects and the predicate signifies the aspects or traits of resources, which indicates relationship between subjects and objects. This technique of describing resources is considerable breakthrough towards Semantic Web roadmap of the W3C's, which is a revolutionary platform of the World Wide Web enabling software systems to store, understand, exchange, and reuse machine-readable information at the internet [5].

Furthermore, W3C has initiated collaborations with the e-Governments to envision the idea of web of linked data for official documents. In past, majority of government data have usually been maintained in tabular form such as tab-separated values (TSV) file format [6]. In order to realize the W3C's mission, namely, "to build and reinforce the community of general public who utilize or encourage the utility of W3C applied techniques to enhance e-Government", it is necessary to concert legacy TSV data into RDF data model, which is the enabling technology for the Semantic Web [7].

A TSV file is a simple format of data storage in tabular form e.g. MS Excel spreadsheet files and database tables. Each line in the table represents one record of the file. Tab character is used as separator between each attribute value of records. The TSV file format has remained amongst more general and widely adopted data storage structure because of its simplicity and manageability [8].

Despite contemporary developments in publishing structured data on the Web, transparent conversion from TSV to RDF and its seamless integration on the web for e-Government and general public data is still a question mark [9]. Inspired by this research gap, we have worked on the generation mechanism for RDF data model from TSV file format. We have not only implemented the direct mapping from TSV to RDF. Furthermore, we have also realized RDF mapping language i.e. RML for the subject purpose. RML is the superset of R2RML, which is W3C's recommended mapping language. R2RML was meant to map the data to RDF data model from relational databases. While, RML has broadened the R2RML's capability and extended its scope to map data from other formats as well.

We have developed a software tool called TSV2RDF for generating RDF data model from TSV file format using direct mapping as well as RML. It is capable to take raw TSV data as input, map it either directly to RDF or generate RML rules specific to that particular data and convert it into RDF triples as output. Generated data model and RML rules are further parsed and visualized as RDF graphs.

This research paper is further organized as follows. The 'Related Work' section presents state-of-the-art developments in the area of TSV to RDF conversion. 'Methodology' presents the proposed design framework along with illustrations. The section 'Results and Discussion' contains the experimental results discussed in detail. 'Conclusions' summarizes the main concept and defines outcomes.

2. Related Work

In this section presented about the existing literature developed for sematic web are illustrated. The analysis is based on categorization of review related to CSV file format.

In [10] proposed an open-source tool designed for translating MS Excel spreadsheet data to RDF. Also in [11] introduced Sheet2RDF platform for the transformation of spreadsheets to RDF data, which enhanced the level of automation. Also, in [12], a method to parse CSV file was presented, in which CSV file is annotated with metadata, tabular data model is generated and then converted into RDF triples. An application designed for transforming and visualizing tabular data from various data portals to RDF model, which supported an incremental mapping from heterogeneous sources [12]. Similarly, in [13] focused on extracting underlying concepts and their interrelations from a set of CSV files. The files are treated as particular concepts and organized into specific domain, referred as domain ontology. Further, the domain ontology is utilized for expressing CSV data and presented in RDF data model.

Based on consideration of different data format in [14] described RML as the RDF mapping language aimed at expressing custom defined rules to map data from heterogeneous data-structures such as TSV, CSV, JSON or XML to RDF. RML is the super ordinate of W3C's recommended R2RML, aimed at extending its application, broadening its horizon and ranging its support for other formats of data. Furthermore, in [15], a novel approach has been described for mapping diversified data structures and hierarchical sources of data to RDF using the RML. Also, in [16] demonstrated and implemented a design that illustrated how to perform scalable and seamless transformation of semantic data.

To process the dataset the Semantic Web, enabling machine-readability for web contents, has the very potential required for the revolution of the World Wide Web. In [17] investigated the various candidate methods that may be used to add semantics with the contents, semantic APIs, tagging and proposed a framework for the assessment of semantic associated with the web of data. The RDF (the Resource Description Framework) for expressing information on the Web is evaluated in [18]. They defined abstract syntax of a data model that serves as linking of all RDF-based specifications and languages. The particular syntax has a set of subject-predicate-object termed as triples as the basic data structures, where the underlying elements may be data-typed literals, blank nodes and/ or IRIs. Further, RDF is the ultimate universal

description language for expressing sematic information at the Web. A text-based syntax was defined for RDF termed in [19].

In [20] designed semantically deep analysis along with disambiguating word sense, recognizing named entity and vocabularies of supervised Semantic Web in an effort of extracting named entities along with their interrelations, from text and them converting them to RDF representation. In [21] is a software tool that is designed to work with Jena framework. Battle, discussed bidirectional mapping in-between XML and RDF. Rather than basing on any mapping language, author used XML schema for the needful. A generic approach for transforming XML data to RDF model in a way that was ontology-dependent is developed in [22]. Tab-separated values (TSV) are a frequently used formalism to represent linguistically annotated natural language, e.g., in the longstanding series of Shared Tasks of the Conference of Natural Language Learning (CoNLL), recent initiatives on the creation of corpora and tools with cross-linguistically applicable ("universal") annotations [Universal Dependencies, UD], [UniMorph], [Universal Propositions], or in computational lexicography and corpus linguistics [Corpus Workbench], [Sketch Engine]. Many such "CoNLL" formats exist, but although they share a number of common features (e.g., one word per line, empty line to mark sentence breaks, comments after #), they are not interoperable with each other, as different pieces of information are represented differently in different dialects, e.g., placed in different columns or spread over multiple columns in one format, but consolidated into one in another. In [23] stated that CoNLL-RDF is a set of tools introduced to facilitate processing and transforming CoNLL and other TSV formats in a serialization-independent way: On the basis of a user-provided mapping from columns to labels (properties), sentence by sentence (blocks of annotations separated by empty lines), tab separated data is transformed to RDF graphs in accordance with the CoNLLRDF data model. Annotations can then be manipulated using SPARQL Update operations and serialized in TSV, RDF or XML formats. Unlike CSV2RDF [23], R2RML [23], and related general-purpose technology for mapping tabular data to RDF, CoNLL-RDF provides linguistic data structures: The CoNLL-RDF data model uses the NLP Interchange Format [NIF] to encode sentences, words and sequential relations between these, and extends it with properties for the annotation of words, syntactic dependencies and semantic roles. First introduced in 2017, this technology is now being used in a number of projects in NLP [23], knowledge engineering [23], linguistics [23] and Digital Humanities [23].

We consulted [18] for RDF graph visualization, in which Antoniazzi & Viola conducted a survey of the prominent tools for the triples' graphical visualization exploiting RDF graph representation. PGV (Paged Graph Visualization) in [19] is an effective semi-autonomous software tool for RDF data visualization. There are two sub components of PGV, which are PGV explorer and RDF pager, respectively.

In [24] addressed various centralized methods for managing and processing RDF data. Four groups of approaches, i.e. partitioning-based approaches, cloud-based solutions, partial evaluation-based approach and federated SPARQL assessment systems, are analyzed in distributed RDF management. In some cloud-based distributed frameworks, such as SHARD [24], H2RDF [24], etc., the storage technique used partitioning-based distributed frameworks WARP [24], TriAD [24]; Distributed Federated Systems, i.e. Distributed systems such as gStore etc. have been tested based on DARQ, SPLENDID etc. and partial question evaluation. The authors do not provide a thorough comparison of these frameworks and there has also been no discussion of distributed query processing in these systems. Only some key approaches were illustrated by Özsu.

In addition to these, in [25], SHAPE [25] etc. are some of the other distributed stores discussed. The authors also analyze the RDF datasets of benchmarks such as LUBM, SP2Bench etc. This study does not address many distributed RDF stores, no specifics are given about indexing and querying in distributed RDF stores, and the authors do not compare these stores. In [26] focused on the distributed RDF systems and thus conduct a detailed experimental assessment of 12 of these systems, i.e. SHARD [26], Form [26] and so on Start-up cost, query performance, scalability and adaptability are the metrics used to measure these systems. These systems are classified based on their model of execution, based on MapReduce and Graph-based technique in RDF. With RDF system partitioning approach exhibits significant performance in terms of storage, indexing, partitioning and framework retrieval. In [27] discussed only some distributed RDF systems H2RDF+ [27] etc. The analysis is based on the comparison of RDF framework with consideration of partitioning, query, storage and indexing. There is no identification of the drawbacks of these systems and the comparison is limited to only a few distributed frameworks addressed.

In [28] Some Semantic Web repositories have been reviewedRDF-3x, BigOWLIM, Jena and Sesame. With significant solutions for RDF data storage and recovery and its challenges. Many distributed RDF systems rely on key value stores, such as Rya [28], CumulusRDF [28], AMADA [28], Stratustore [28], H2RDF [28] and MAPSIN [28]. have also been reviewed. These structures are not addressed in detail by the authors and do not demonstrate their drawbacks. These structures are addressed only briefly and the significant elements such as partitioning and indexing are not closely examined. The authors also only use the MapReduce processing framework to address the Major RDF frameworks and do not analyze the other related systems based on the Spark framework. To overcome those limitation and improve processing time this paper proposed a software for

conversion of TSV file to RDF format.

3. Preliminaries

This section presented the description about mapping technique involved in conversion of file format. In this paper TSV2RDF software is developed for conversion of TSV file in RDF using mapping technique. The general description about mapping technique is evaluated.

3.1 Direct Mapping

Two approaches to data processing are RDF and Graph Databases. Modelling, storing and querying graph-like data is based on that. The database systems based on these models are becoming increasingly important in the field. They are applied in different application domains where complicated data analytics are used.

3.1.1 Database Mapping

Generally, mapping a database is a tool for translating databases. From a model of source databases to a model of target databases. We should take into account Two types of database mapping: direct database mapping, which allows for direct mapping of databases Automatic database translation without any user feedback; Database mappings and manuals, which include additional details (e.g., an ontology) for carrying out the database.

3.1.2 Direct Mapping Schema and Instances

Consider M as a database model with schema of M with instances A. The database instances of M with ordered pair of $D_M = (S_M, I_M)$ in which schema is represented as S_M and instances as $I_{M.}$ For the database schema instances I_M and schema denoted as S_M, in which I_M is validated with variable S_M represented as $I_{M=}|S_M$. Once I_M is satisfied database $D_M = (S_M, I_M)$ with validated if condition $I_{M=}|S_M$ is satisfied.

3.2 RML Mapping

As the purpose of RML is to support heterogeneous data, it is important to support sources and data sources in different formats. The certain formulation, data in a specific format, which may be route and query or custom grammar languages. For instance, Data can be referred to via XPath in an XML file and to a relational database Via SQL. The Reference Formulation, to this end, The Reference Formulation was added to this end, suggesting the formulation used in a certain data source to refer to data.

Although R2RML handles mapping data definitions effectively to RDF in relational databases, there is no standardized mapping language to support other formats. RML is described as a superset aimed at extending its applicability and extending its reach beyond tabular structures. In heterogeneous formats, and define mappings of data. We briefly introduce R2RML in this section, address its constraints because of its assumption of a tabular input, and explain how RML extends R2RML to handle hierarchical structures.

RML syntax is specification with mapping for extraction of scheme for expression of targeted language (rml:referenceFormulation). The value expression are extracted from the target source with RDF Term Map or iterator (rml:iterator). In triple map expression need to be corrected with Triple Map expression with assured accuracy (rml:referenceFormulation). The processor of RML is expressed with modular architecture with module extraction and mapping with each other with ensemble expression. The module of RML mapping is processed with execution of mapping with RML syntax description. In extraction module language is expressed with specified execution for expression of return values. The function is therefore limited to parsing the given source and mapping of data extraction as stated.

4. Proposed Model Framework

We have implemented two different approaches to generate RDF data model from TSV file i.e. direct mapping and RML based mapping. The block diagram of proposed methodology is presented at Figure1. The underlying semantics and detailed narrative of both approaches is described below.

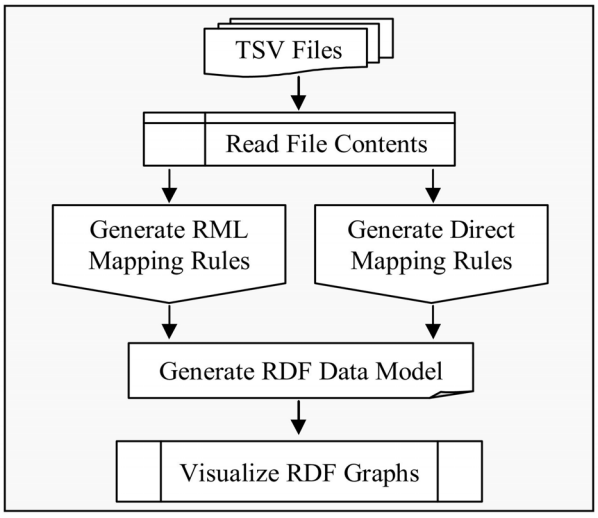

Figure 1. TSV2RDF Block Diagram

The proposed TSV2RDF involved in conversion of TSV files in to RDF format. The proposed model processes the TSV files and perform Mapping rules. The mapping technique utilized for conversion of file format are RML mapping and Direct mapping involved in identification of variables. The proposed model utilizes two mapping models RML and direct mapping. Initially RML mapping is performed followed by Direct mapping technique. The process involved in RML and Direct mapping are presented as follows:

4.1 RML based Mapping

The first technique is based on RML i.e. RDF mapping language. W3C has initially recommended R2RML as

No.	RML	R2RML
1	Logical Source (e.g. TSV, CSV, JSON, XML etc.)	Logical Table (e.g. RDMS table)
2	URI (Source Pointer)	Name of the Table
3	Reference	Column
4	Reference Formulation	SQL
5	Defined Iterator	Iteration per Row

Table 1. RML Extensions over R2RML

the mapping language for linking data to RDF model from relational databases. RML is introduced as superset of R2RML, which has extended R2RML capability and broadened its scope to support mappings of data from other formats e.g. TSV, CSV, XML, HTML and JSON as well. The value additions provided by RML over R2RML are summarized in Table 1.

RML has extended R2RML logical table and introduced logical source which need not only be the relational database table but of any data format. Table name of R2RML has been replaced by IRI pointer to the source by RML. RML has reference attribute in place of column for R2RML. RML has introduced the property named reference formulation to clarify which data input format is required to be parsed. Furthermore, as R2RML tables had implicit iteration mechanism for each row, there was a requirement of defined explicit iterator for RML as it needs to process the input data from heterogeneous formats having no iteration pattern.

We have implemented the proposed methodology in Microsoft Visual Studio 2012, using C# 5.0 based on .NET Framework 4.5. The developed software is named as TSV2RDF. The screenshot of the software is presented in Figure 2. The algorithm for RML based mapping is shown below.

Algorithm: RML based Mapping

1. Input: CSV File

2. Output: RDF Data Model

3. Processing

3.1: Read the TSV file contents

3.2: Set IRI (Internationalized Resource Identifier)

3.3: Define schema name

3.4: Describe prefixes

3.5: Generate subjects, predicates and objects

4. Modeling:

4.1: Generate the RDF data model

4.2: Visualize the RDF graph

We demonstrate step-by-step execution of the implemented methodology with the help of an example.

A. Step 1: Read the TSV File Contents

TSV2RDF takes a TSV file as input, which is required to be provided by pressing the "Browse" button and navigating to the file path. Once an appropriate file is selected, its contents are displayed at the software interface, as shown in Figure 2.

Our example input file contains the basic information about four different persons. The column names are ID, Name, Gender and Country. We are interested to annotate the information of every person and then generate the equivalent RDF triples. We will explain what RML rules are needed to generate the required RDF triples and how we write them.

B. Step 2: Define RML Mapping Rules

Basically, 2x set of rules are required for RML based RDF generation.

• Rules for describing the input TSV file.

• Rules for defining the RDF terms generation mechanism from TSV file and their usage phenomena.

4.2 Direct Mapping

Our second methodology has been established on direct mapping from TSV data to RDF. A parser has been developed which reads the contents of the TSV file and produces a set of RDF triples for each row of the file. The algorithm implemented for direct mapping is shown below.

Algorithm: Direct Mapping

1. Input: CSV File

2. Output: RDF Data Model

3. Processing

3.1: Read the TSV file contents

3.2: Define prefixes

3.3: Generate Subject IRI

3.4: Generate predicates

3.5: Generate objects

4. Modeling:

4.1: Generate the RDF data model

4.2: Visualize the RDF graph

The subject IRI is generated by concatenating the three components i.e. the IRI title, column name and its value. The predicate is formed for each column by concatenating the IRI title with the column name. The RDF literals values are formed by accessing the lexical form of the column value.

The row identifier is set as the object for generated triples. Step-by-step execution of the implemented direct mapping based RDF generation mechanism is demonstrated with the help of the same TSV example file which was previously used in RML based RDF generation technique.

Step 1: Read the TSV File Contents

As, TSV2RDF takes a TSV file as input, which has already been discussed in previous section of RML based mapping. We will consider the same example file to explain the direct mapping technique to generate the required RDF triples.

Step 2: Define Direct Mapping Rules

4.3 Example for RML and Direct Mapping

In RML mappling particular example we will need following rules:

• IRI representing each person by concatenating "http://example.org/IRI Title/" with the person ID, which will be used as subject for the triples.

• A person will be annotated with class schema:Person

• Name will be annotated with property schema:Name

• Gender will be annotated with property schema:Gender

• Country will be annotated with property schema:Country

1) Define Prefixes

RML mapping rules start with defining the prefixes. Table II represents the essential prefixes required to be defined.

RDF classes and properties will be defined using schema and dbo prefix. The software interface for defining prefixes is shown in Figure 2. Default values for every control have been provided with edit facility. User can customize the prefixes values as per requirement. User defined values of prefixes will be used to generate RML mapping rules and RDF triples.

The prefixes finalized by the user will be added in the Turtle document as follows:

```
@prefix rml:<http:/ semweb.mmlab.be/ns rml#>
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix csvw:<http://www.w3.org/ns/csvw#>
@prefi rr:<http:// www.w3.org/ns/r2rml#>
@prefix ql:<http://semweb.mmlab.be/ ns/ql#>
@prefix :<http://example.org/rules/>
@prefix dbo:<http://dbpedia.org/ontology/>
@prefix schema:<http://schema.org/>
```

2) Map Input TSV File

RML mapping rule to specify which particular TSV file is required to be mapped, is added as follows:

```
:TriplesMap a rr:TriplesMap;
rml:logicalSource [
rml:source [
    a csvw:Table;
csvw:url "<fileName.tsv>";
csvw:dialect [
     a csvw:Dialect;
csvw:delimiter "\t"
   ]
  ];
rml:referenceFormulationql:CSV
 ].
```

Element-wise illustration of the rule is as follows:

No.	Prefix	Description
1	RML	Ontology of RML
2	RDF	Vocabulary of RDF related concepts
3	CSVW	Vocabulary of CSVW, used to describe TSV file
4	RR	Ontology of R2RML, augmented by RML
5	QL	Vocabulary of query processing language
6	Empty	Prefix defined for our custom RML based rules
7	DBO	Ontology of DBpedia
8	Schema	Vocabulary of schema.org

Table 2. RML Prefixes

• *":TriplesMap a rr:TriplesMap;"* is used to define the Triples Map to group all the rules.

• *":TriplesMaprml:logicalSource [ ]"* is used to contain all the rules specific to input TSV format file along with methods to read data from that file.

• *"rml:source [ ]"* is used to contain all the rules about the source TSV file.

• *"csvw:url<**Persons.tsv**>"* identifies our input TSV file.

• *"csvw:dialect [csvw:delimiter**"\t"**]"* specifies that the tab character i.e. \t is used as delimiter.

• *"rml:referenceFormulationql:CSV"* specifies that column names will be used to access data from TSV file.

3) Generate Subjects

Subject, IRI title and schema are defined using respective controls at software interface as shown in Fig. 2. RML rule to generate subject IRI with title is added as follows:

```
:TriplesMaprr:subjectMap [
   rr:template "http://example.org/<IRItitle>/<{sub
   ject}>"
].
```

Element-wise illustration of the rule is as follows:

":TriplesMaprr:subjectMap [ ]" is used to contain all the rules regarding subject of the triple.

*"rr:template "http://example.org/**Persons**/{ID}"* specifies that our subject IRI is generated with the concatenation of http://example.org/IRI Title/ with the data value of the ID column from our specific input TSV file.

4) Generate Predicates and Objects

Class Annotations: RML rules to annotate every row of the input TSV file with the schema class is added as follows:

```
:TriplesMaprr:predicateObjectMap [
rr:predicaterdf:type;
rr:constant schema:<schemaName>
].
```

Element-wise illustration of the rule is as follows:

• *":TriplesMaprr:predicateObjectMap [ ]"* is used to contain entire rules about particular predicate for triple.

• *"rr:predicaterdf:type"* specifies type of the predicate.

• *"rr:objectMap [ ]"* is used to contains all rules about specific object for a triple.

• *"rr:constantschema:**TSV**"* specifies schema:TSV is the object for every triple of our specific input TSV file.

Property Annotations: RML rules to annotate every column (other than Subject IRI) of the input TSV file with the schema, are added as follows:

```
:TriplesMaprr:predicateObjectMap [
rr:predicate schema:<columnName>;
rr:objectMap [
rml:reference<"columnName">
 ]
].
```

These rules are a bit dissimilar from the ones to annotate a class. *rr:constant* is replaced with *rml:reference* because of the different object for every row. More specifically, *"[rml:reference "**Name**"]*, *"[rml:reference "**Gender**"]* and *"[rml:reference "**Country**"]* specify that the data of the column "Name", "Gender" and "Country" are referred for the objects of our specific input TSV file.

C. Step 3: Generate the RDF Data Model

Once input TSV file specific RML mapping rules are defined, they are executed to generate RDF triples. In our specific example following triples are generated.

```
@prefix      dbo:<http://dbpedia.org/ontology/> .
@prefix      schema:<http://schema.org/> .
<http://example.org/Persons/1> a schema:TSV;
schema:Name "M Hasan";
schema:Gender "Male";
schema:Country "Azerbaijan".
<http://example.org/Persons/2> a schema:TSV;
schema:Name "Y Li";
schema:Gender "Female";
schema:Country "China".
<http://example.org/Persons/3> a schema:TSV;
schema:Name "Z M Ma";
schema:Gender "Male";
schema:Country "China".
<http://example.org/Persons/4> a schema:TSV;
schema:Name "M Waqas";
schema:Gender "Male";
schema:Country "Pakistan".
```

Four triples have been generated: one for each row i.e. each person, coupled with unique subject IRI and annotated with class schema:TSV.

D. Step 4: Visualize the RDF Graph

We have subscribed the "RDF Grapher", which is an online web service for RDF data parsing and graph visualization. The defined RML mapping rules are listed in Figure 2. and generated RML based RDF triples are shown at Figure 3.

Direct Mapping
After completion of RDF mapping this research perform direct mapping those examples are stated as follows:

Define Prefixes
Direct mapping rules start with defining the prefixes. Table 3 represents the essential prefixes required to be defined.

No.	Prefix	Description
1	Base	The base IRI (Internationalized Resource Identifier)
2	XSD	The XML schema definition

Table 3. Direct Mapping Prefixes

The prefixes will be added in Turtle document as follows:

@base <http://example.org/>

@prefix xsd:<http://www.w3.org/2001/XMLSchema#>

5) Generate Subject IRI
The Subject and IRI title have already been defined using respective controls at software interface as shown in Fig. 2. Our subject IRI is generated by the concatenation of title/subject with the value of the column ID from our specific input TSV file i.e. <Persons/ID=1> to <Persons/ID=4>.

6) Generate Predicates
The predicate for each column is composed by joining the IRI title along with the column name like <Persons#ID>, <Persons#Name>,<Persons#Gender> and <Persons#Country>inour test example.

7) Generate Objects
The object is picked by applying grid search at the tabular data. Column headings are matched with the row identifiers to reach at object value. For example *'M Hasan'* is selected as object for the subject IRI *<Persons/ID=1>* and the predicate *<Persons#Name>*. Similarly*'Male'* is identified as object for the subject *<Persons/ID=1>* and predicate *<Persons#Gender>*. Furthermore *'Azerbaijan'* is denoted as object for subject *<Persons/ID=1>* and predicate *<Persons#Country>*.

Step 3: Generate the RDF Data Model

After defining the input TSV file specific direct mapping rules, they are executed to generate RDF triples. In our test case example, following triples are generated.

@base <http://example.org/>
@prefix xsd:<http://www.w3.org/2001/XMLSchema#>
<Persons/ID=1><Persons#ID> '1' .
<Persons/ID=1><Persons#Name> 'M Hasan' .
<Persons/ID=1><Persons#Gender> 'Male' .
<Persons/ID=1><Persons#Country> 'Azerbaijan' .
<Persons/ID=2><Persons#ID> '2' .
<Persons/ID=2><Persons#Name> 'Y Li' .
<Persons/ID=2><Persons#Gender> 'Female' .
<Persons/ID=2><Persons#Country> 'China' .
<Persons/ID=3><Persons#ID> '3' .
<Persons/ID=3><Persons#Name> 'Z M Ma' .
<Persons/ID=3><Persons#Gender> 'Male' .
<Persons/ID=3><Persons#Country> 'China' .
<Persons/ID=4><Persons#ID> '4' .
<Persons/ID=4><Persons#Name> 'M Waqas' .
<Persons/ID=4><Persons#Gender> 'Male' .
<Persons/ID=4><Persons#Country> 'Pakistan' .

Step 4: Visualize the RDF Graph

The result of direct mapping for TSV data at "RDF Grapher" is shown at Figure 4.

5. Experimental Analysis

In this section presented about the experimental setup and dataset considered for analysis are presented.

5.1 Experimental Setup
Performance of the developed TSV2RDF software has been evaluated by transforming ten benchmark TSV datasets to RDF. The experiments have been performed on Intel® Core i5-3570 CPU 3.40 GHz with 4GB RAM. The publicly available datasets have been downloaded from well reputed data service providers including UCI machine learning repository, GitHub, Kaggle, Eurostat and IMDb.

5.2 Dataset Description
Internet movies database (IMDb): IMDb is famous collection of online datasets related to films and television programs. Each dataset is a TSV file. We have selected a representative dataset named "title.ratings.tsv", which contains the information about IMDb rating and votes for movies and television program titles.

Online videos data at UCI machine learning repository: The UCI machine learning repository is one of the most prestigious source of databases and domain specific theories. The online videos data has been selected for evaluating the performance of our work. This dataset contains more than a million randomly sampled and compiled video instances listing for ten essential video characteristics presented along with the video ID from YouTube.

NASA access log 1 at Kaggle: Kaggle, a subsidiary of the Google LLC, lies among the world's largest dataset

providers. NASA access log dataset has been selected from Kaggle to showcase our performance. It is comprised of two log files which contain two months' worth of entire HTTP user requests to the Florida based NASA Kennedy Space Center's server.

NASA access log 2 at Kaggle: The second file from NASA access log at Kaggle repository has also been downloaded to evaluate test performance of TSV2RDF. It is another representation of the previously described dataset. It contains further trace for NASA web server access log.

ADW dataset at GitHub: GitHub Inc. is a US based global company and subsidiary of Microsoft Inc., that provides hosting service for software development. ADW dataset has been dowaded from Github at Pilehvar/ADW.

Aact_ali01 dataset at Eurostat: Eurostat is the statistical data processing office of the European Union to provide accurate, high quality and realistic statistics for Europe. Aact_uv01 is the TSV database of Economic accounts for agriculture, which is the national reference metadata.

Aact_eaa01 dataset at Eurostat: Aact_eaa01 is another TSV database downloaded from Eurostat's bulk download facility.

Aact_uv01 dataset at Eurostat: Aact_uv01 is the 3rd TSV file downloaded from Eurostat statistical data of European Union.

Labeled train data at Kaggle: The labeled train data is the user sentiment analysis data downloaded from Kaggle.

GAP coreference data at GitHub: GAP is the gender-balanced database containing various coreference-labeled ambiguous pronoun and antecedent name pairs, originally sampled from Wikipedia and eventually released by Google.

6. Simulation Analysis

The proposed TSV2RDF is implemented with selected dataset for analysis of processing time. The proposed TSV2RDF involved in processing of TSV files with different mapping approaches such as RML and direct mapping. With application of RML and direct mapping TSV files are processed and converted. The processed data and its attributes are presented in table 4 as follows. TSV2RDF has successfully converted all the selected TSV datasets to RDF triples. Further specifications of the datasets along with TSV2RDF performance for RDF generation is highlighted in Table 4.

TSV2RDF can't be compared with the existing online and offline tools as it is unique of its kind. Existing RDF generation tools and techniques either don't support TSV as the input file format or not based on RML. In Figure 2 the proposed TSV2 RMD is presented. The analysis is based on the consideration of four schema such as ID, name, gender and country. In Figure 2 for schema 'ID' proposed TSV2RMD is displayed and in Figure 3 RML mapping for ID is illustrated and Figure 4 provides the schema 'ID' direct mapping is presented.

Figure 3 provides the RDF mapping for schema 'ID' and in figure 4 provides the direct mapping for schema. The analysis exhibited that triples mapping involved in estimation of all the schema for analysis. In figure 5 RML mapping for schema 'Name' is illustrated and figure 6 provides triples direct mapping for schema 'Name' is presented.

Dataset	Source	Size MB	Colu-mns	No. of Rows	Process- ing Time (Sec)
Internet Movies	IMDb	16.7	3	1030009	70
YouTube Videos	UCI	8.5	20	168286	20
NASA Access Log	Kaggle	38.3	9	500000	85
NASA Access Log	Kaggle	39.5	9	531958	90
ADW dataset	GitHub	0.12	3	65	0
Aact_ali01 dataset	Eurostat	0.35	38	120	0
Aact_eaa01 dataset	Eurostat	14	38	53853	35
Aact_uv01 dataset	Eurostat	0.95	38	4888	2
Labeled Train Data	Kaggle	32	3	25000	50
GAP dataset	GitHub	1.03	11	2000	1

Table 4. IMDB Dataset Details

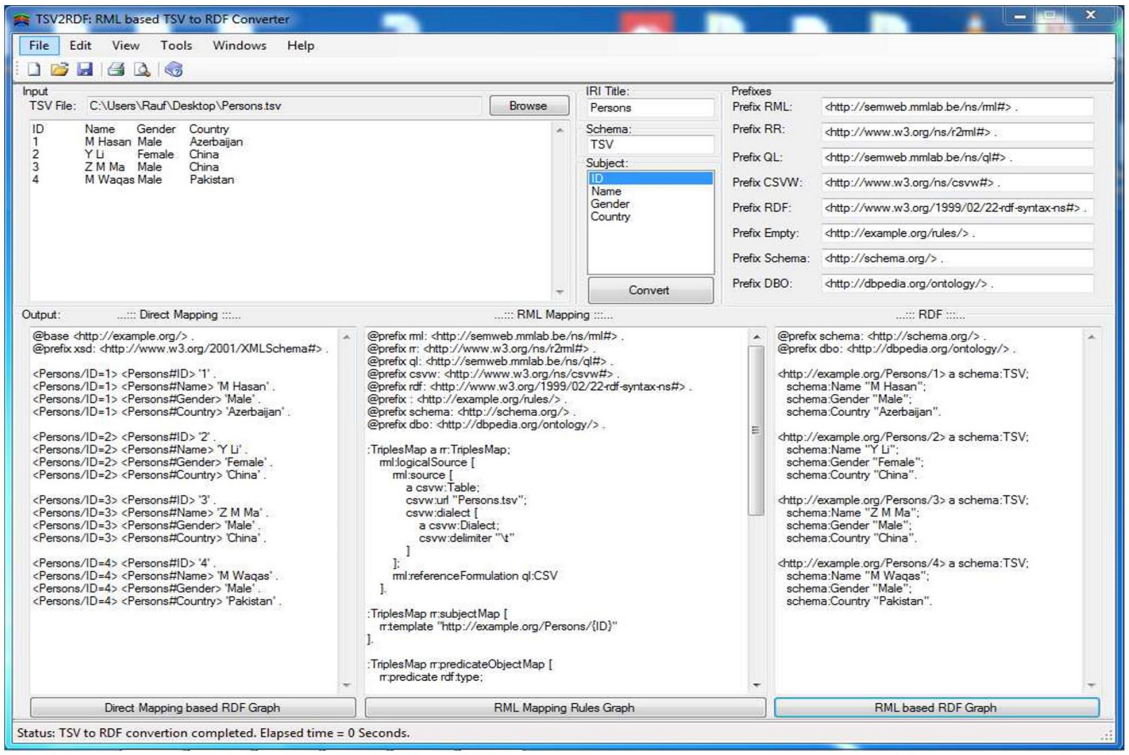

Figure 2. TSV2RDF Software for Schema 'ID'

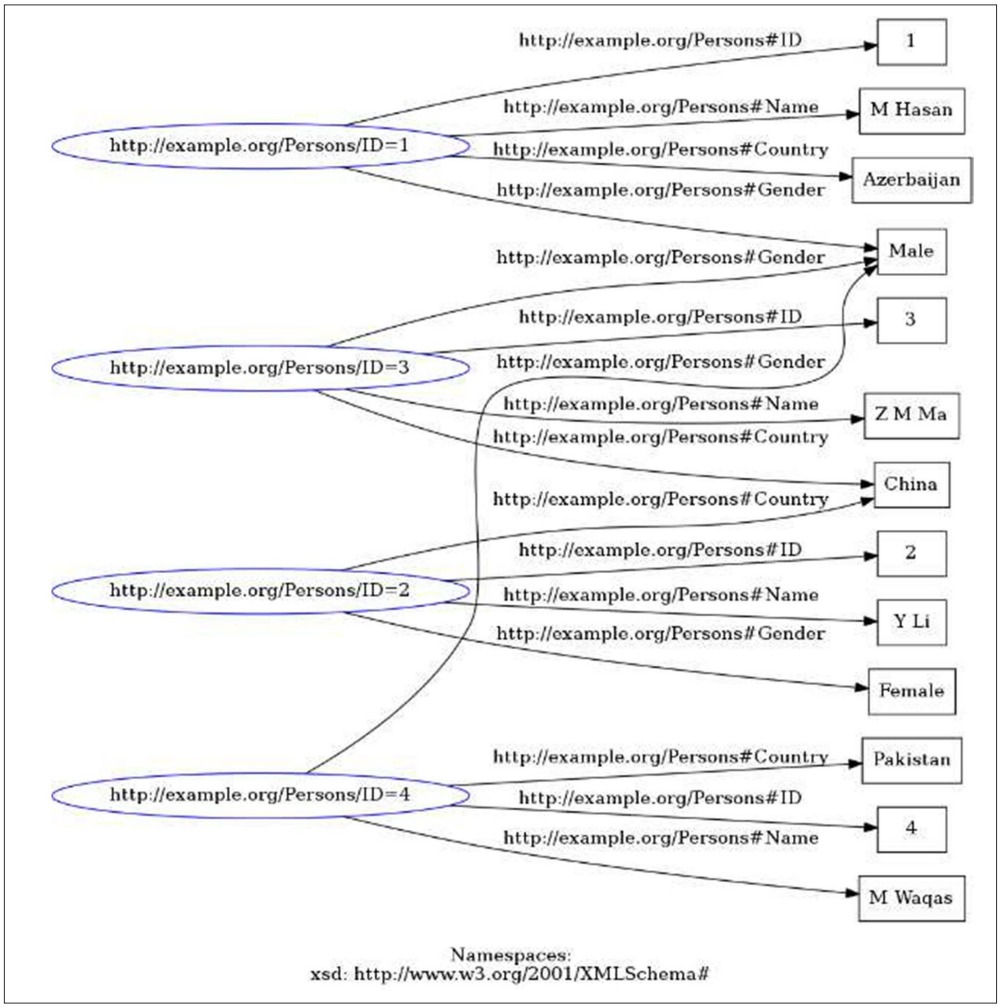

Figure 3. RML Mapping based RDF Triples

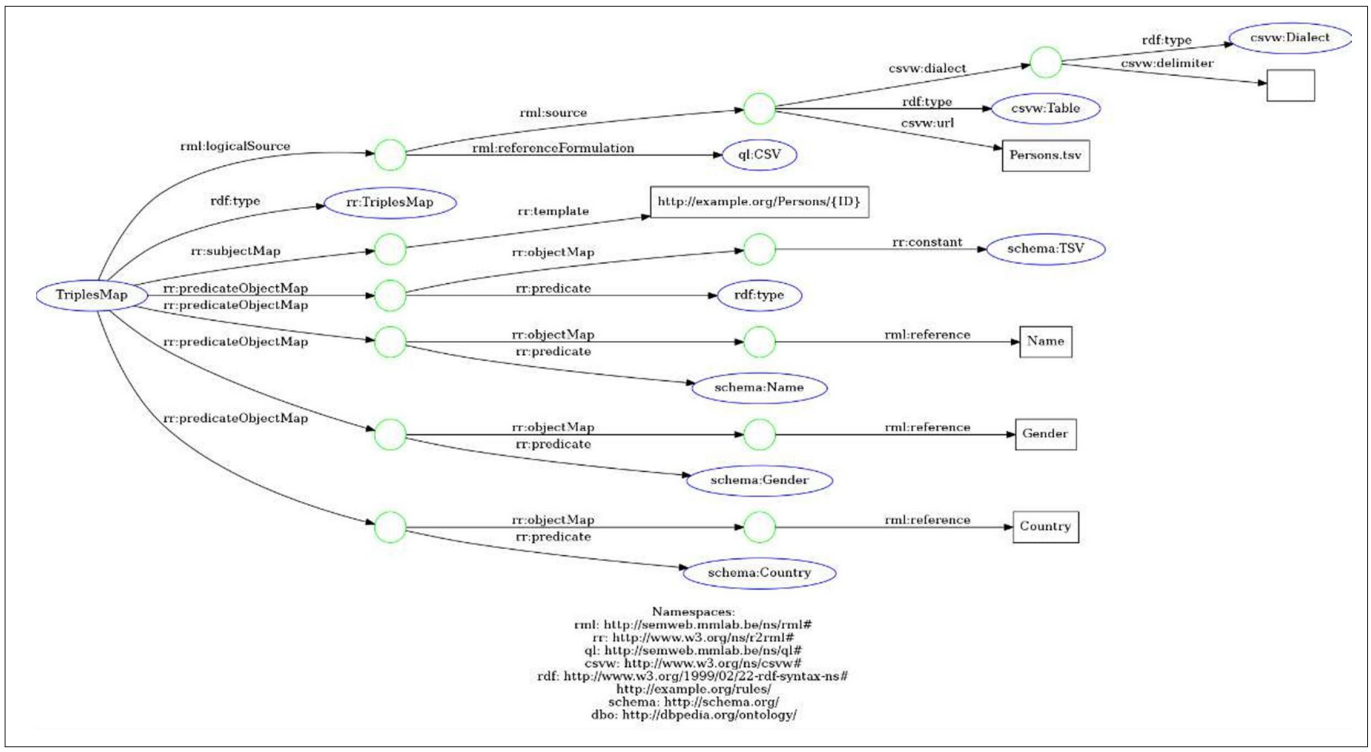

Figure 4. Direct Mapping based RDF Triples

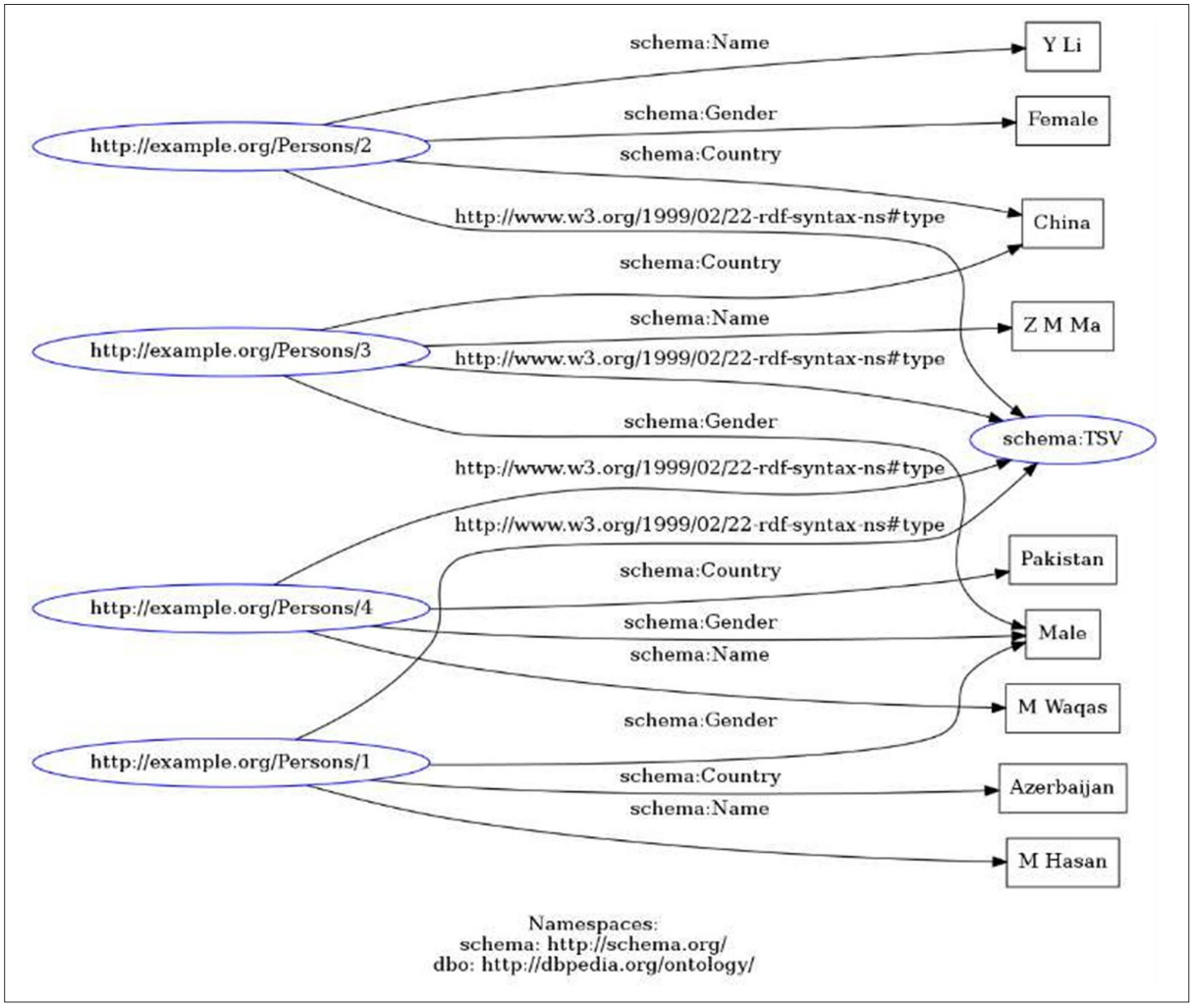

Figure 5. RML mapping for Schema 'Name'

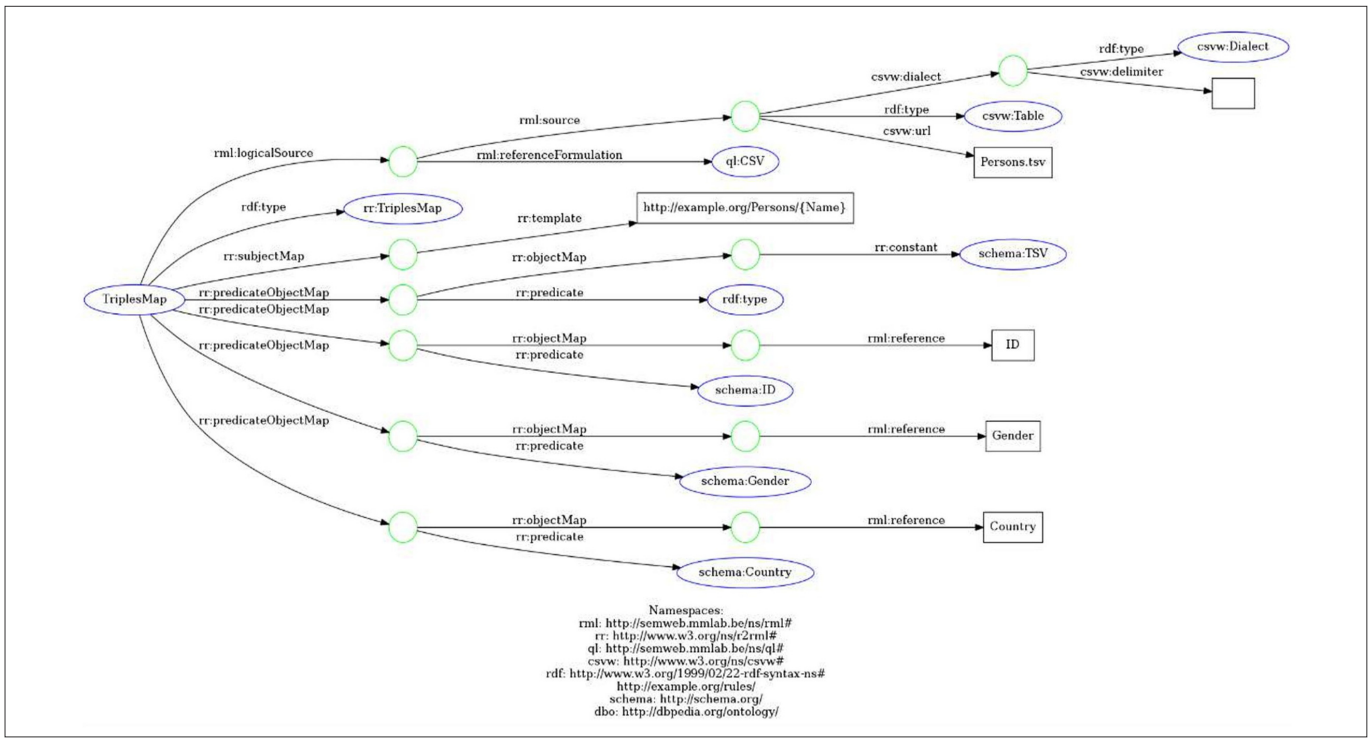

Figure 6. Direct Mapping Triples for Schema 'Name'

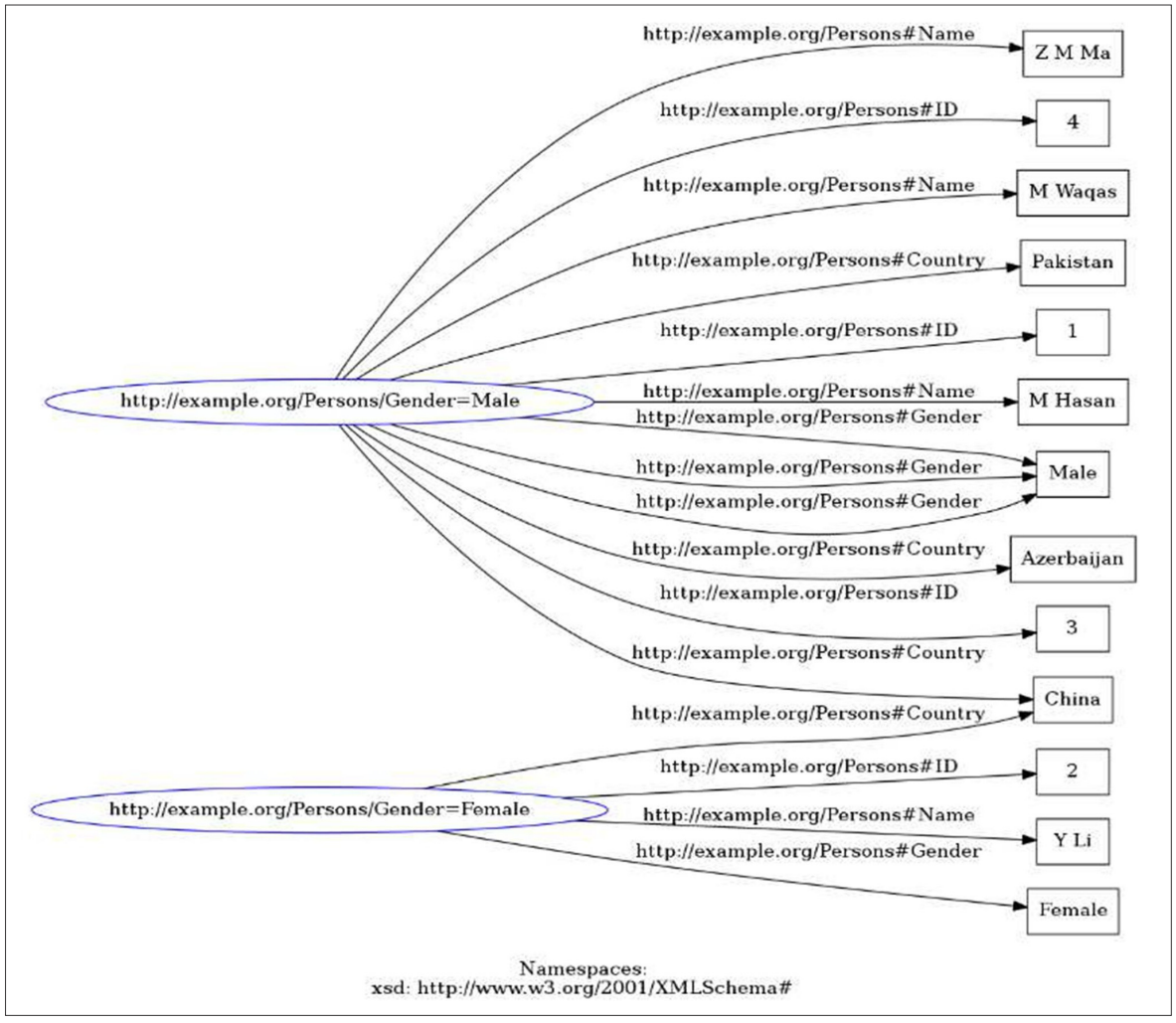

Figure 7: RML Mapping for Schema 'Gender'

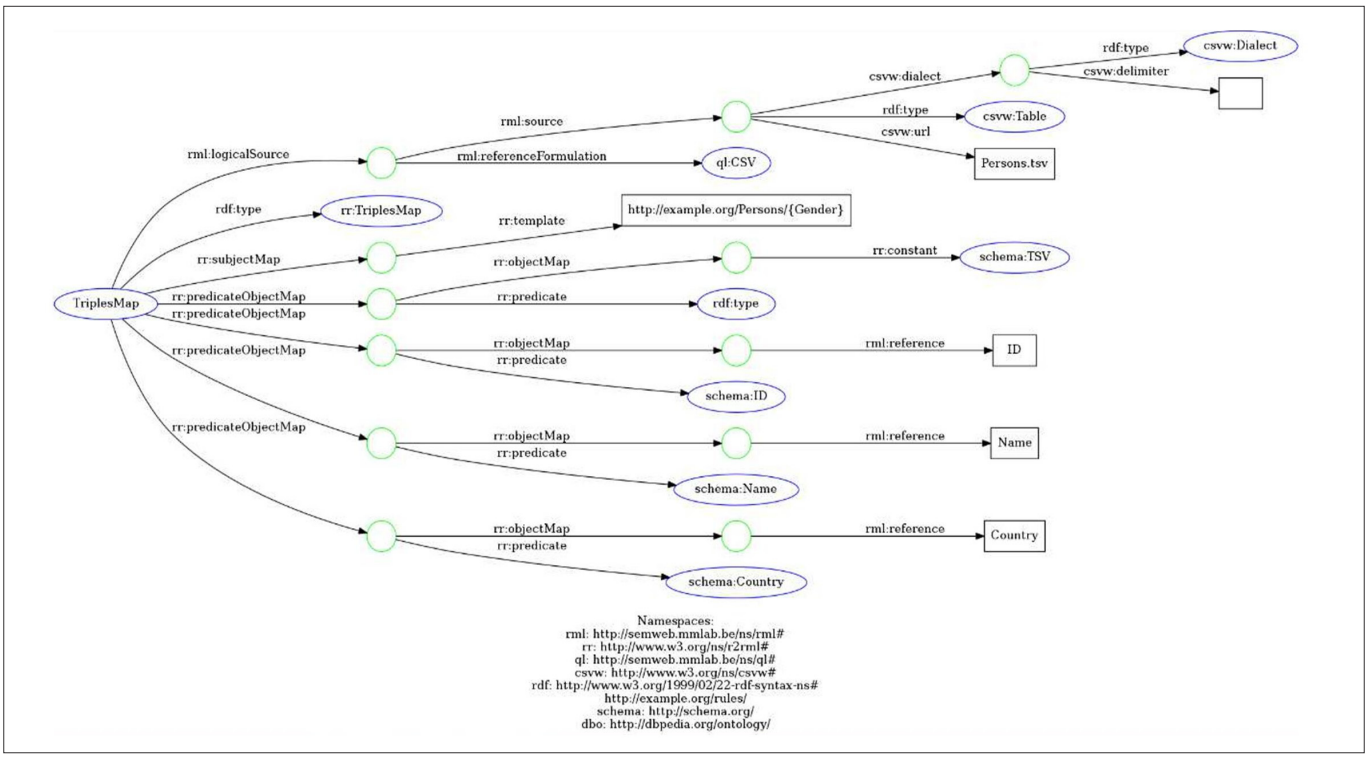

Figure 8. Direct Mapping for Schema 'gender'

As gender selected are two variables male and female which involved in processing of schema. In Figure 9 and Figure 10 RML and direct mapping for schema 'Country' is illustrated.

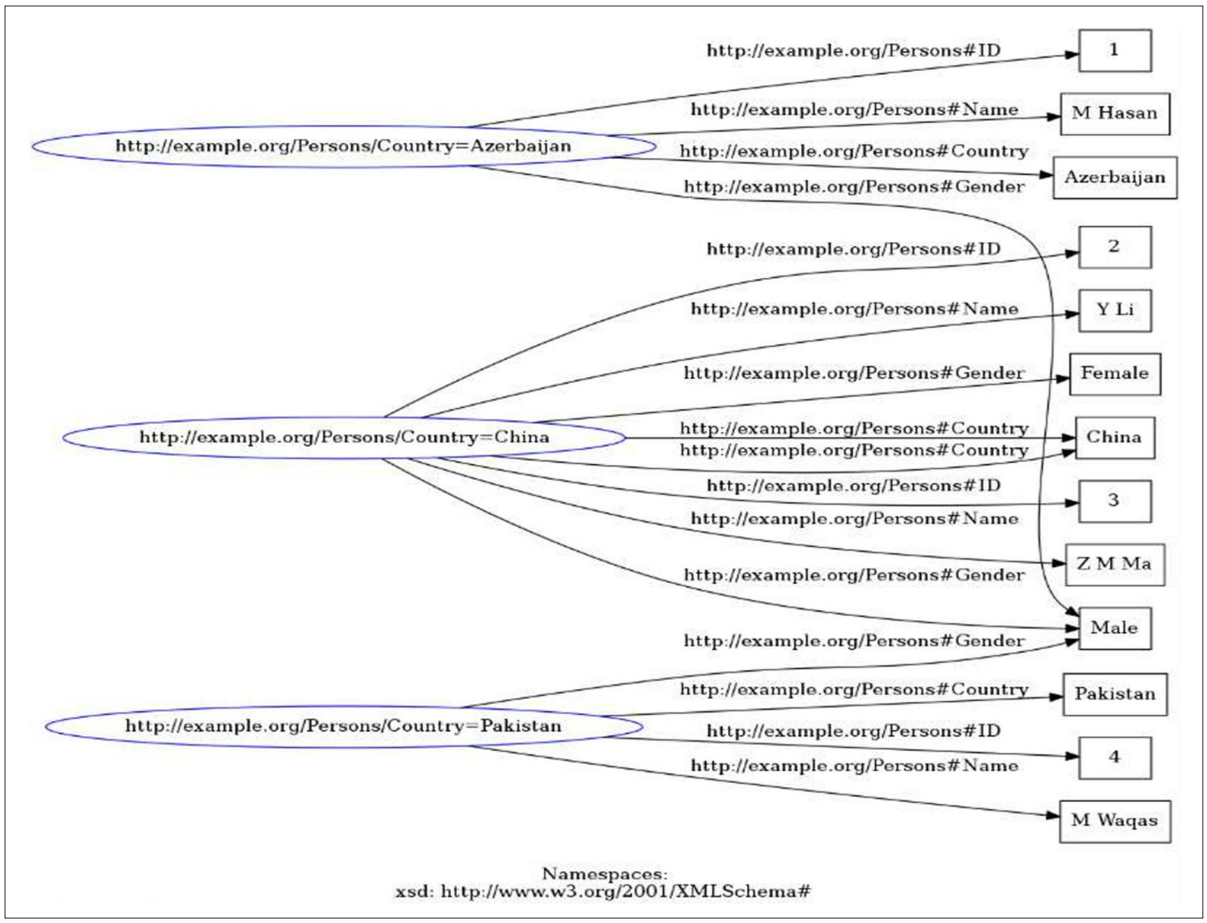

Figure 9. RML Mapping for Schema "Country"

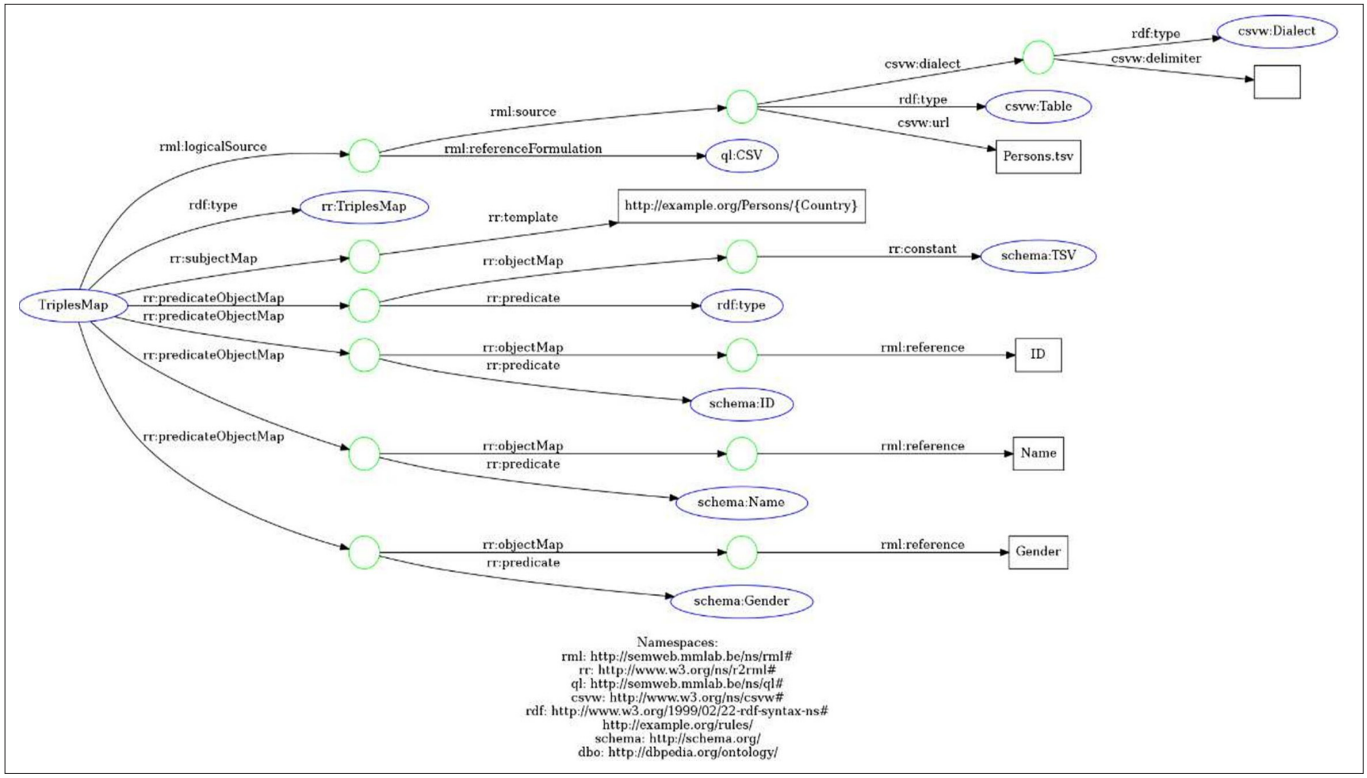

Figure 10. Direct Mapping for Schema 'Country'

Dataset	Size MB	Processing Time (Sec)
DBpEn	62M	90
DBpNL	21M	106
DBpAII	-	95
DBLP	12M	120
iLastic	150K	70
CDFLG	0.6K	140
CEUR-WS	2.4K	90

Table 5. Comparison of Processing Time

When it comes to RDF generation for semantic web, TSV has suffered the dilemma of least studied file format. Furthermore, the rare existed TSV to RDF transformation studies have either implemented the direct mapping or based on R2RML. While, direct mapping is usually restricted to conventions, which lacks the standardization and as R2RML is specialized language meant for linking relational data sources to RDF, which constraint it from being general purpose mapping language for heterogeneous data formats. We have realized this research gap and utilized the capabilities of RDF mapping language i.e. RML to introduce a novel method of generating RDF data model from TSV file format, which will prove to be a major contribution in this particular field. Thus, our TSV2RDF is a remarkable innovation, built on Semantic Web technologies.

Based on the proposed TSV2RMD process the existing dataset for analysis are presented in table 5.

The analysis of processing time (sec) existing dataset considered for analysis are presented. The dataset considered for analysis exhibited that CDFLG exhibits higher processing time of 140 seconds and iLastic processing time is minimal with 70 secs. In figure 11 comparative analysis of processing time for different dataset are presented.

In Table 6 processing time estimated for proposed TSV2RMD average time is estimated for proposed and existing technique dataset were presented. In figure 12 processing time estimated for proposed and existing technique is presented.

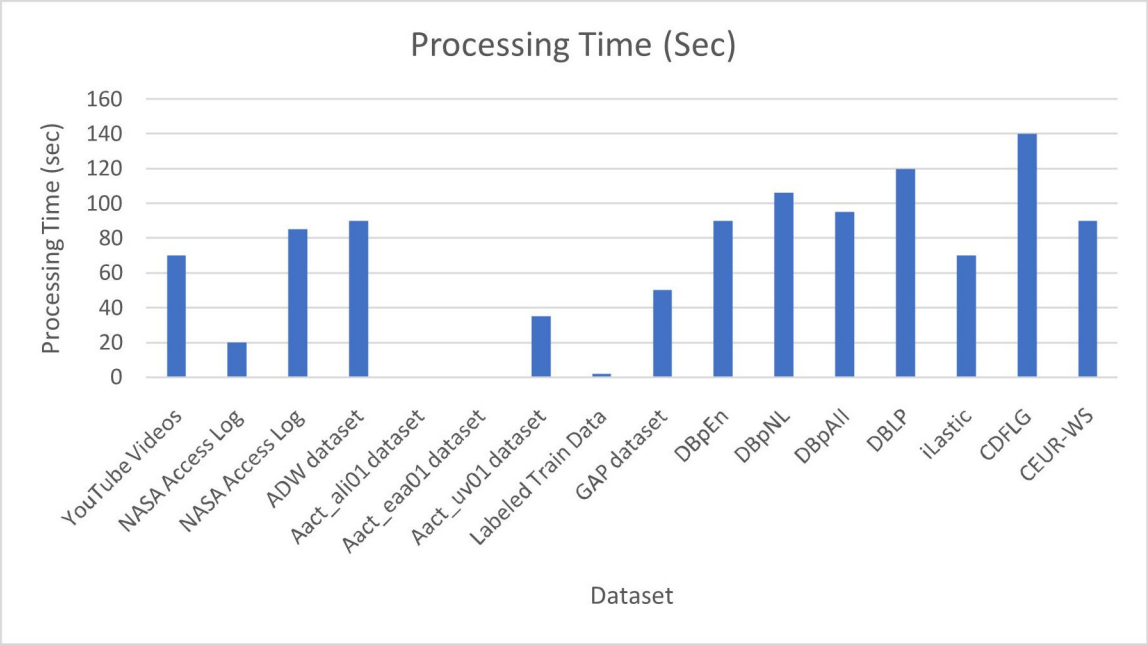

Figure 11. Comparison of Processing Time

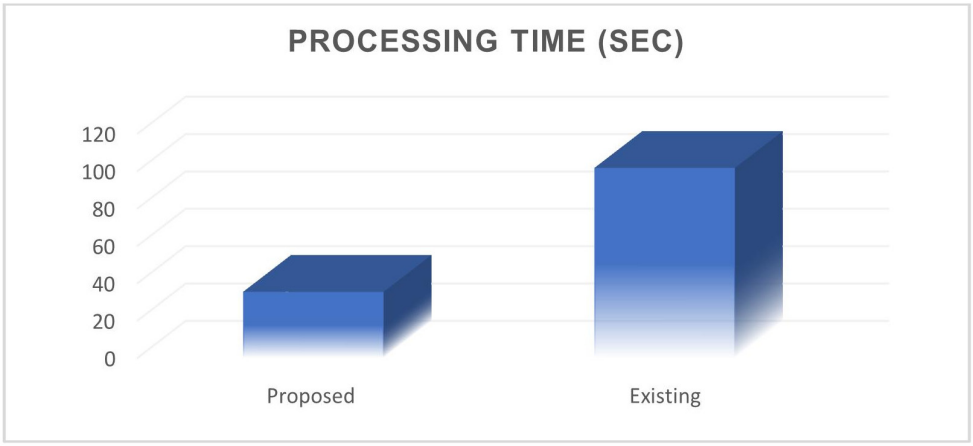

Figure 12. Processing Time with existing and proposed

Dataset	Processing Time (Sec)
Proposed	35.3
Existing	101.6

Table 6. Processing Time

The comparative analysis of processing time exhibited that average processing time of existing technique is observed as 101.6 sec and proposed technique exhibits minimal processing time of 35.3 sec. The analysis of results stated that proposed TSV2RMD significantly reduces the processing time rather than existing technique.

7. Conclusion and Future Work

The Semantic Web technology is meant to enhance computers capability at better understanding and processing the information. For the purpose, RDF has been evolved as the de facto standard for associating precise meanings to the data. However, there is dire need to convert legacy data of majority Governments and other officials from heterogeneous file formats to RDF for the sake of standardization among diversities. Furthermore, this transformation is required to be based on well-established standards. As RML is emerging as state-of-the-art RDF mapping language and it supports numerous data formats, we have developed a software TSV2RDF for generating RDF data model from TSV file using RML as well as direct mapping. A case study has also been presented illustrating the working of our software. TSV2RDF has been tested at a collection of representative TSV datasets from industry. It has generated the accurate RDF triples with less processing time. TSV2RDF is suggested as an effective value addition at TSV to RDF data model transformation with engaging usability.

Realizing the strength of RML and attaining a decent grasp on the working principles of RML rules, we are also interested to work on RML based RDF data model generation from various other least studied but very common legacy file formats like TXT, RTF and text-based DAT.

References

[1] Wagner, A., Bonduel, M., Pauwels, P., Rüppel, U. (2020). Representing construction-related geometry in a semantic web context: A review of approaches. Automation *in Construction*, 115, 103130.

[2] Dadkhah, M., Araban, S., Paydar, S. (2020). A systematic literature review on semantic web enabled software testing. *Journal of Systems and Software*, 162, 110485.

[3] Banane, M., Belangour, A., El Houssine, L. (2017, October). Storing RDF data into big data NoSQL databases. *In*: First International Conference on Real Time Intelligent Systems (p. 69-78). Springer, Cham.

[4] Faqir, A., Mahmood, A., Qazi, K., Malik, S. (2019, November). An Approach to Map Geography Mark-up Language Data to Resource Description Framework Schema. *In*: International Conference on Intelligent Technologies and Applications (p. 343-354). Springer, Singapore.

[5] Elbashir, M. K., Aboelhassan, M. A. (2018). An Algorithm for Mapping Relational Database to Resource Description Framework. *Gezira Journal of Engineering and Applied Sciences*, 11 (1).

[6] de Paula, G. C., de Farias, C. R. (2020). A competency question-oriented approach for the transformation of semi-structured bioinformatics data into linked open data. *Engineering Applications of Artificial Intelligence*, 90, 103495.

[7] Lefrançois, M., Zimmermann, A., Bakerally, N. (2017, May). A SPARQL extension for generating RDF from heterogeneous formats. *In*: European Semantic Web Conference (pp. 35-50). Springer, Cham.

[8] Karr, J. R., Liebermeister, W., Goldberg, A. P., Sekar, J. A., & Shaikh, B. (2020). Structured spreadsheets with ObjTables enable data reuse and integration. *arXiv preprint arXiv:2005.05227*.

[9] Chiarcos, C., Ionov, M. (2019). Ligt: An LLOD-native vocabulary for representing interlinear glossed text as RDF. *In*: 2nd Conference on Language, Data and Knowledge (LDK 2019). Schloss Dagstuhl-Leibniz-ZentrumfuerInformatik.

[10] Dotsika, F. (2010). Semantic APIs: Scaling up towards the semantic web. *International Journal of Information Management*, 30 (4) 335-342.

[11] Miller, E. (1998). An introduction to the resource description framework. *Bulletin of the American Society for Information Science and Technology*, 25 (1) 15-19.

[12] Lefrançois, M., Zimmermann, A., Bakerally, N. (2016, November). Flexible RDF generation from RDF and heterogeneous data sources with SPARQL-Generate. In *European Knowledge Acquisition Workshop* (p. 131-135). Springer, Cham.

[13] Chiarcos, C., Ionov, M., Glaser, L., Fäth, C. (2020). An ontology for CoNLL-RDF: Formal data structures for TSV formats in language technology.

[14] Barisevièius, G., Coste, M., Geleta, D., Juric, D., Khodadadi, M., Stoilos, G., Zaihrayeu, I. (2018, October). Supporting digital healthcare services using semantic web technologies. *In*: International Semantic Web Conference (p. 291-306). Springer, Cham.

[15] Wang, X., Zhang, X., Li, M. (2015). A survey on semantic sensor web: sensor ontology, mapping and query. *International Journal of u-and e-Service, Science and Technology*, 8 (10) 325-342.

[16] Schwarz, J., Terrenghi, N., Legner, C. (2017). Towards comparable business model concepts: resource description framework (RDF) schemas for semantic business model representations. *In*: *Designing the Digital Transformation: DESRIST 2017 Research in Progress Proceedings of the 12th International Conference on Design Science Research in Information Systems and Technology. Karlsruhe, Germany. 30 May-1 Jun.* (p. 101-109). KarlsruherInstitutfürTechnologie (KIT).

[17] Lin, Z., Tripunitara, M. (2017, March). Graph Automorphism-Based, Semantics-Preserving Security for the Resource Description Framework (RDF). *In*: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy (p. 337-348).

[18] Liu, J., Yang, M., Zhang, L., Zhou, W. (2019). An effective biomedical data migration tool from resource description framework to JSON. *Database*.

[19] Fang, H., Zhao, B., Zhang, X. W., Yang, X. X. (2019). A united framework for large-scale resource description framework stream processing. *Journal of Computer Science and Technology*, 34 (4) 762-774.

[20] Hadi, A. S., Ali, S. H. (2019). Resource Description Framework Representation for Transaction Log File. *Journal of Computational and Theoretical Nanoscience*, 16 (3) 1093-1099.

[21] Faheem, M., Sattar, H., Bajwa, I. S., Akbar, W. (2018, October). Relational database to resource description framework and its schema. *In*: International Conference on Intelligent Technologies and Applications (p. 604-617). Springer, Singapore.

[22] Mandal, K., Sen, T. (2018). *U.S. Patent No. 10,042,619*. Washington, DC: U.S. Patent and Trademark Office.

[23] Matsumoto, S., Yamanaka, R., Chiba, H. (2018). Mapping RDF graphs to property graphs. *arXiv preprint arXiv:1812.01801*.

[24] Lin, Z., Tripunitara, M. (2017, March). Graph Automorphism-Based, Semantics-Preserving Security for the Resource Description Framework (RDF). *In*: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy (p. 337-348).

[25] Riaz, A., Bajwa, I. S., Ali, M. (2019, November). Automatic RDF, Metadata Generation from Legacy Software Models. *In*: International Conference on Intelligent Technologies and Applications (p. 385-397). Springer, Singapore.

[26] De Una, D., Rümmele, N., Gange, G., Schachte, P., & Stuckey, P. J. (2018, January). Machine Learning and Constraint Programming for Relational-To-Ontology Schema Mapping. *In*: *IJCAI* (Vol. 2018, p. 27th).

[27] Dingman, P. C., Bunton, W. G., Van Dyken, K. E., Yogman, L. T., Zhang, Y. (2018). *U.S. Patent No. 10,127,250*. Washington, DC: U.S. Patent and Trademark Office.

[28] Cate, B. T., Kolaitis, P. G., Qian, K., Tan, W. C. (2017). Approximation algorithms for schema-mapping discovery from data examples. *ACM Transactions on Database Systems (TODS)*, 42 (2) 1-41.