From Engineering To Research

The KG Construction Use Case

David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain

Samaneh Jozashoori, SDM-TIB
Maria-Esther Vidal, SDM-TIB
Enrique Iglesias, University of Bonn
Diego Collarana, Fraunhofer IAIS
Oscar Corcho, OEG-UPM

✉ dchaves@fi.upm.es

dchavesf

23/07/2020

Online

DO NOT QUOTE ME ON THIS PRESENTATION

Thoughts of a junior researcher after ~4 years working on Semantic Data Integration

Motivation I - Reviews

“... **not a precise**, full methodology that others can u SWC2019)

“... transformation process **should be** a particular implementation-...” (ISWC2019)

“... focused on t nslation over data in CSV formats ...” (

“... I'm not con **an be applied systematically** ...” (ESWC2019)

“... Although there **t much scientific value**, I believe this paper to be of good quality and quite valuable as systems paper ...” (ESWC2020)

1) We were not doing BASIC research
2) Is this a research problem?

Motivation II - Potential Users

“I prefer to develop my **own ad-hoc script** for creating the K ”
→ Ask Ana Iglesias about current status of the Bio2R

“Mappings are really **difficult to unders** nual effort to create the mappings is very high”
→ More than programmin

“Your solutio
→ What is yo ce their methods

“Your **engines a** cult to use/…”
→ Generalizable c s/workflows/frameworks

1) Lighten the learning curve
2) Propose general research solutions
3) Transference to industry

From THE HOW to THE WHAT

- Inverting too much explaining your code
- Focus on the technology problems
- Specific solutions
- Try to explain ALL what I solved
- “Fight” with your competitors

The How

Not so important
for a research work

- Think before do
- Invest time in defining Research Questions
- Make your proposal general
- Implement the solution in a specific case(s)
- Discuss with your co-authors before doing anything
- Technology is a support not the core
- We are researchers, not engineers
- Good theoretical background of XXX (Databases)

The What

Not easy to find
and define

Two contributions

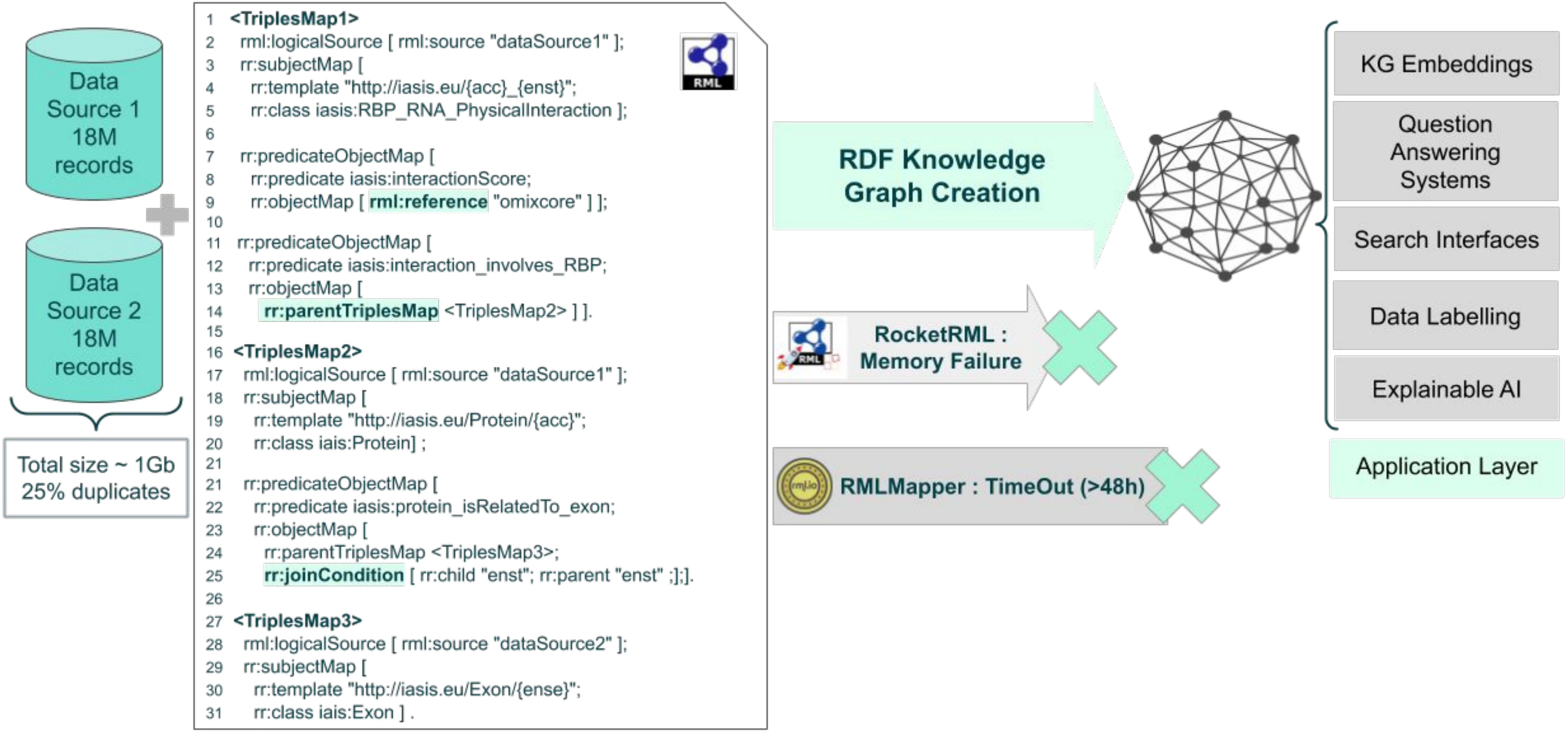
Enrique Iglesias, Samaneh Jozashoori*, David Chaves-Fraga*, Diego Collarana and Maria-Esther Vidal.* **SDM-RDFizer: An RML Interpreter for the Efficient Creation of RDF Knowledge Graphs.** Under review at CIKM20 Resource Track

Samaneh Jozashoori, David Chaves-Fraga*, Enrique Iglesias, Oscar Corcho and Maria-Esther Vidal.* **FunMap: Efficient Execution of Functional Mappings for Scaled-Up Knowledge Graph Creation.** Under review at ISWC20 Research Track

*The authors contributed equally to this research.

SDM-RDFizer: An RML Interpreter for the Efficient Creation of RDF Knowledge Graphs.

SDM-RDFizer - Motivation



Data Source 1 18M records

Data Source 2 18M records

Total size ~ 1Gb
25% duplicates

```
<TriplesMap1>
 rml:logicalSource [ rml:source "dataSource1" ];
 rr:subjectMap [
   rr:template "http://iasis.eu/{acc}_{enst}";
   rr:class iasis:RBP_RNA_PhysicalInteraction ];

 rr:predicateObjectMap [
   rr:predicate iasis:interactionScore;
   rr:objectMap [ rml:reference "omixcore" ] ];

 rr:predicateObjectMap [
   rr:predicate iasis:interaction_involves_RBP;
   rr:objectMap [
     rr:parentTriplesMap <TriplesMap2> ] ].

<TriplesMap2>
 rml:logicalSource [ rml:source "dataSource1" ];
 rr:subjectMap [
   rr:template "http://iasis.eu/Protein/{acc}";
   rr:class iais:Protein] ;

 rr:predicateObjectMap [
   rr:predicate iasis:protein_isRelatedTo_exon;
   rr:objectMap [
     rr:parentTriplesMap <TriplesMap3>;
     rr:joinCondition [ rr:child "enst"; rr:parent "enst" ;];].

<TriplesMap3>
 rml:logicalSource [ rml:source "dataSource2" ];
 rr:subjectMap [
   rr:template "http://iasis.eu/Exon/{ense}";
   rr:class iais:Exon ] .
```

RDF Knowledge Graph Creation

RocketRML : Memory Failure

RMLMapper : TimeOut (>48h)

KG Embeddings

Question Answering Systems

Search Interfaces

Data Labelling

Explainable AI

Application Layer

SDM-RDFizer - Objectives & Research Questions

Problem: Efficient knowledge graph creation in complex data integration scenarios

Objectives:

O1) Define data structures that speed up the execution of mapping rules for KGC

O2) Implement a set of unique physical operators for managing the data structures

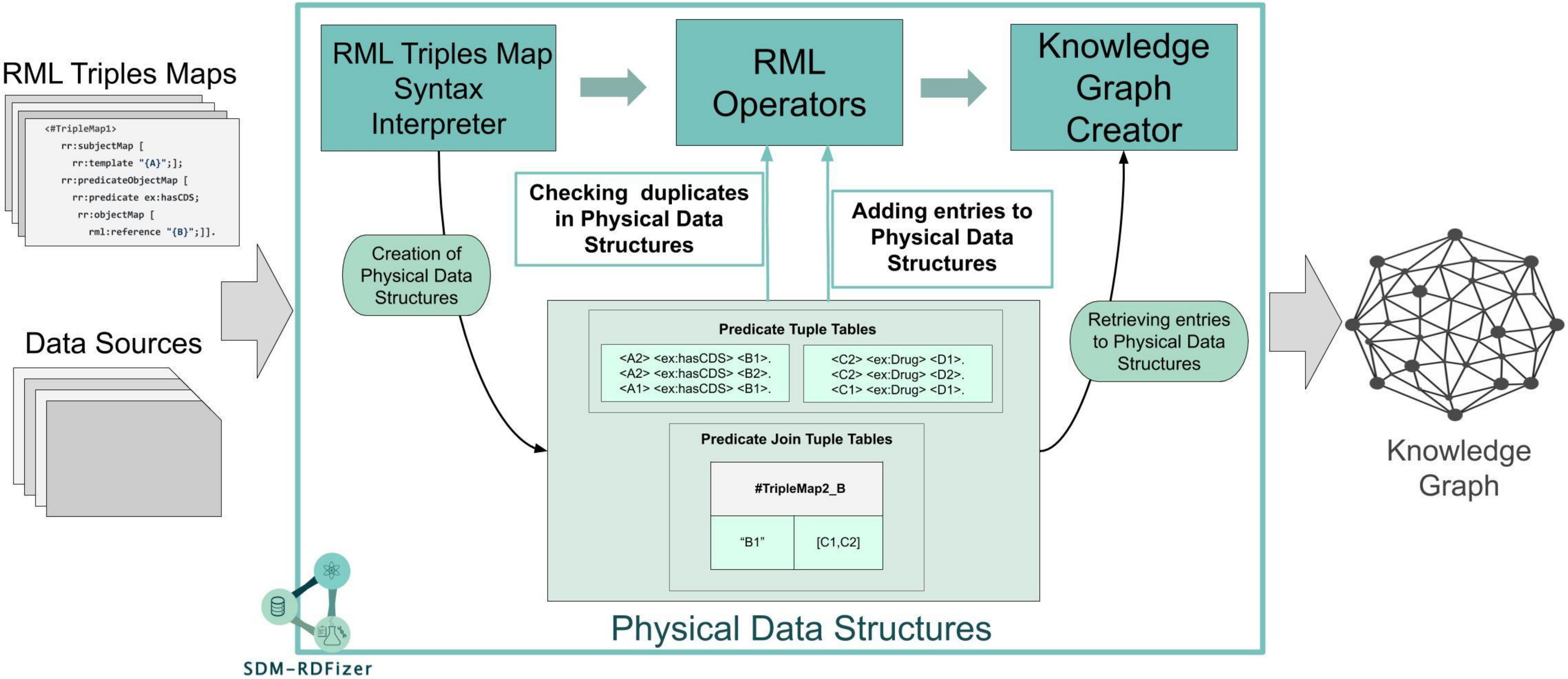
Research Questions:

Q1) What is the impact of data duplication rate in the execution time of a knowledge graph creation approach?

Q2) What is the impact of input data size in the total execution time of a knowledge graph creation process?

Q3) What is the effect of the triples map types in the PredicateObjectMap over the existing engines?

SDM-RDFizer - Architecture



RML Triples Maps
Data Sources
RML Triples Map Syntax Interpreter
RML Operators
Knowledge Graph Creator
Creation of Physical Data Structures
Checking duplicates in Physical Data Structures
Adding entries to Physical Data Structures
Retrieving entries to Physical Data Structures
Physical Data Structures
Knowledge Graph
SDM-RDFizer

```
<#TripleMap1>
    rr:subjectMap [
      rr:template "{A}";];
    rr:predicateObjectMap [
      rr:predicate ex:hasCDS;
       rr:objectMap [
         rml:reference "{B}";]].
```

Predicate Tuple Tables

```
<A2> <ex:hasCDS> <B1>.
<A2> <ex:hasCDS> <B2>.
<A1> <ex:hasCDS> <B1>.
```

```
<C2> <ex:Drug> <D1>.
<C2> <ex:Drug> <D2>.
<C1> <ex:Drug> <D1>.
```

Predicate Join Tuple Tables

#TripleMap2_B	
“B1”	[C1,C2]

SDM-RDFizer - Properties

Number of operations for KGC	Naïve Approach	SDM-RDFizer operators
Simple Object Map	$\|N_p\| + \|S_p\| + \Theta(N_p log(N_p))$	$\|N_p\| + 2\|S_p\|$
Object Reference Map (natural join)	$\|N_p\| + \|S_p\| + \Theta(N_p log(N_p))$	$\|N_p\| + 2\|S_p\|$
Object Join Map (join)	$\|N_{parent}\| \times \|N_{child}\| + \|N_p\| +$ $+ \|S_p\| + \Theta(N_p log(N_p))$	$2\|N_{parent}\| + \|N_{child}\| + \|N_p\| + 2\|S_p\|$

|Np| : Cardinality of a multiset for all RDF triples of p, where p is a predicate (RDF KG with duplicates)
|Sp| : Cardinality of a set all RDF triples of p, where p is a predicate (RDF KG without duplicates)
Θ(.) : Duplicates removal algorithm |Npartent/child|: Columns in parent/child

Experimental results

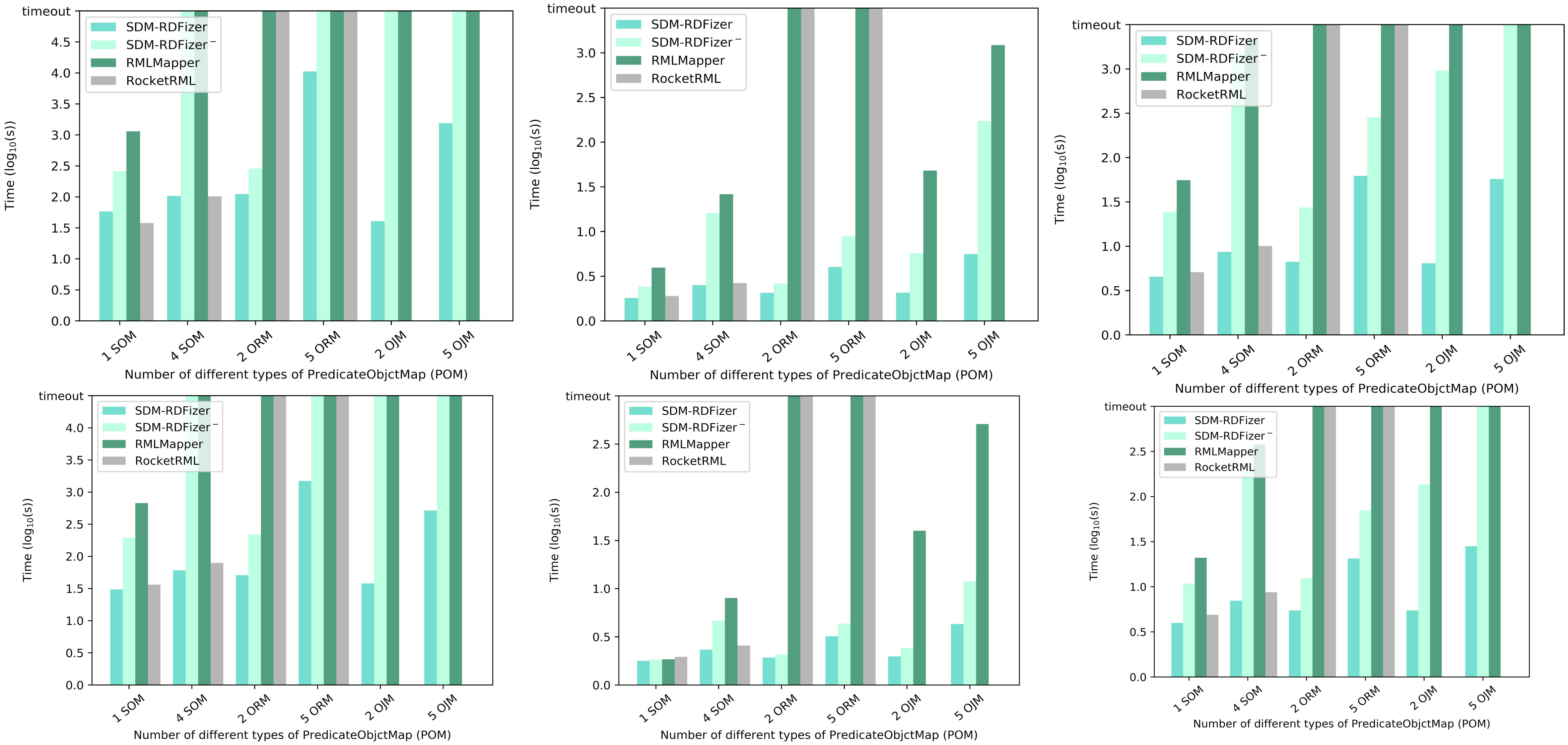
10K rows

100K rows

1M rows

25% Dup

75% Dup

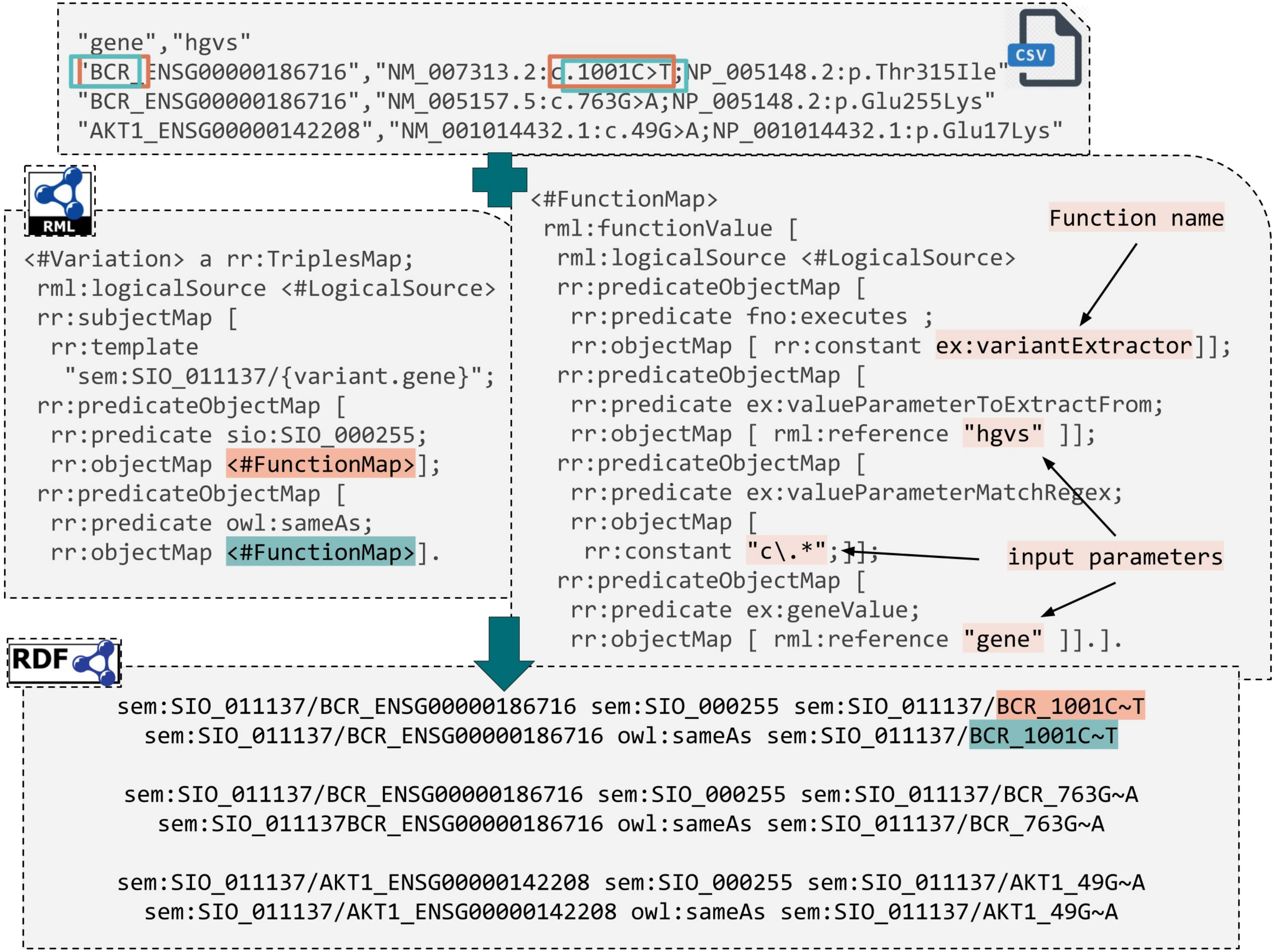


Conclusions

- **Novel physical operators and data structures** that speed up the generation of duplicate-free KG
- Empirical results indicate that SDM-RDFizer **outperforms the state of the art** by up to three orders of magnitude
- Basis for the development of **real-world knowledge graph applications**

FunMap: Efficient Execution of Functional Mappings for Scaled-Up Knowledge Graph Creation

FunMap - Motivation



```
"gene","hgvs"
"BCR_ENSG00000186716","NM_007313.2:c.1001C>T;NP_005148.2:p.Thr315Ile"
"BCR_ENSG00000186716","NM_005157.5:c.763G>A;NP_005148.2:p.Glu255Lys"
"AKT1_ENSG00000142208","NM_001014432.1:c.49G>A;NP_001014432.1:p.Glu17Lys"
```

```
<#Variation> a rr:TriplesMap;
 rml:logicalSource <#LogicalSource>
 rr:subjectMap [
  rr:template
   "sem:SIO_011137/{variant.gene}";
 rr:predicateObjectMap [
  rr:predicate sio:SIO_000255;
  rr:objectMap <#FunctionMap>];
 rr:predicateObjectMap [
  rr:predicate owl:sameAs;
  rr:objectMap <#FunctionMap>].
```

```
<#FunctionMap>
 rml:functionValue [
  rml:logicalSource <#LogicalSource>
  rr:predicateObjectMap [
   rr:predicate fno:executes ;
   rr:objectMap [ rr:constant ex:variantExtractor]];
  rr:predicateObjectMap [
   rr:predicate ex:valueParameterToExtractFrom;
   rr:objectMap [ rml:reference "hgvs" ]];
  rr:predicateObjectMap [
   rr:predicate ex:valueParameterMatchRegex;
   rr:objectMap [
    rr:constant "c\.*";]];
  rr:predicateObjectMap [
   rr:predicate ex:geneValue;
   rr:objectMap [ rml:reference "gene" ]].].
```

```
sem:SIO_011137/BCR_ENSG00000186716 sem:SIO_000255 sem:SIO_011137/BCR_1001C~T
 sem:SIO_011137/BCR_ENSG00000186716 owl:sameAs sem:SIO_011137/BCR_1001C~T

sem:SIO_011137/BCR_ENSG00000186716 sem:SIO_000255 sem:SIO_011137/BCR_763G~A
 sem:SIO_011137BCR_ENSG00000186716 owl:sameAs sem:SIO_011137/BCR_763G~A

sem:SIO_011137/AKT1_ENSG00000142208 sem:SIO_000255 sem:SIO_011137/AKT1_49G~A
 sem:SIO_011137/AKT1_ENSG00000142208 owl:sameAs sem:SIO_011137/AKT1_49G~A
```

FunMap - Objectives & Research Questions

Problem: Scaled-up KG construction from functional mapping rules

Objectives:

O1) Transform a data integration system with functional mappings into an equivalent data integration system where mappings are function-free

O2) Optimization techniques to reduce the total execution time of the KGC

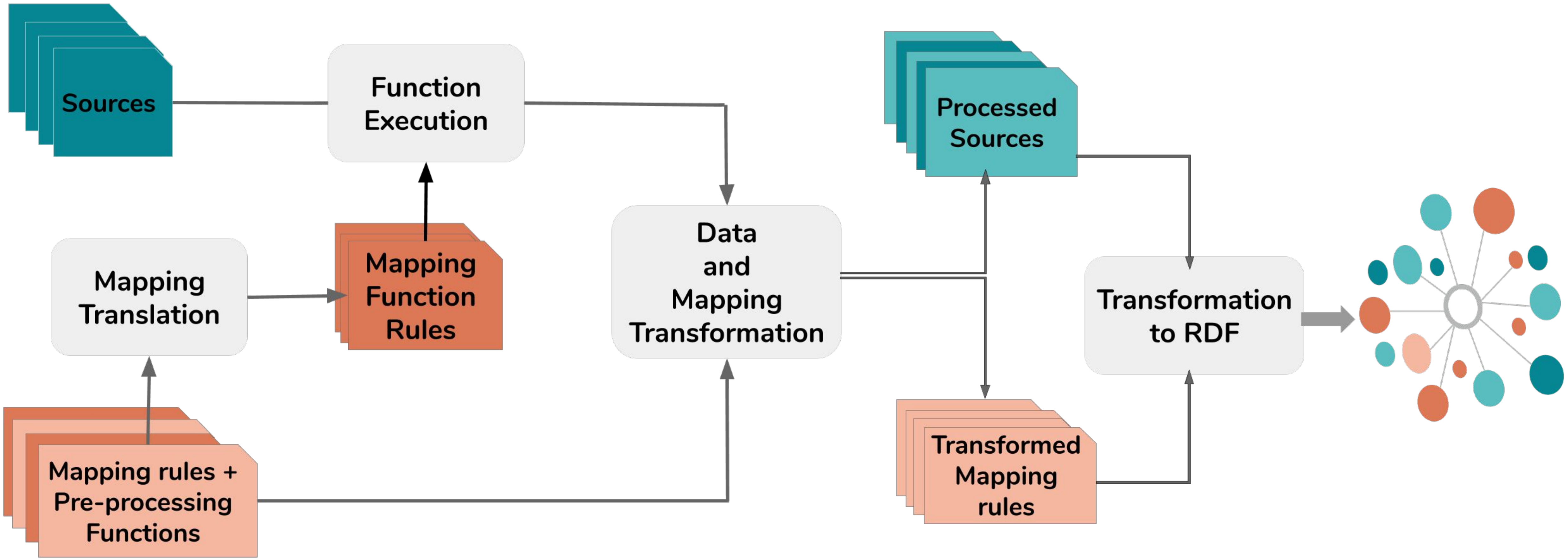
Research Questions:

Q1) What is the impact of data duplication rate in the execution time of a knowledge graph creation approach?

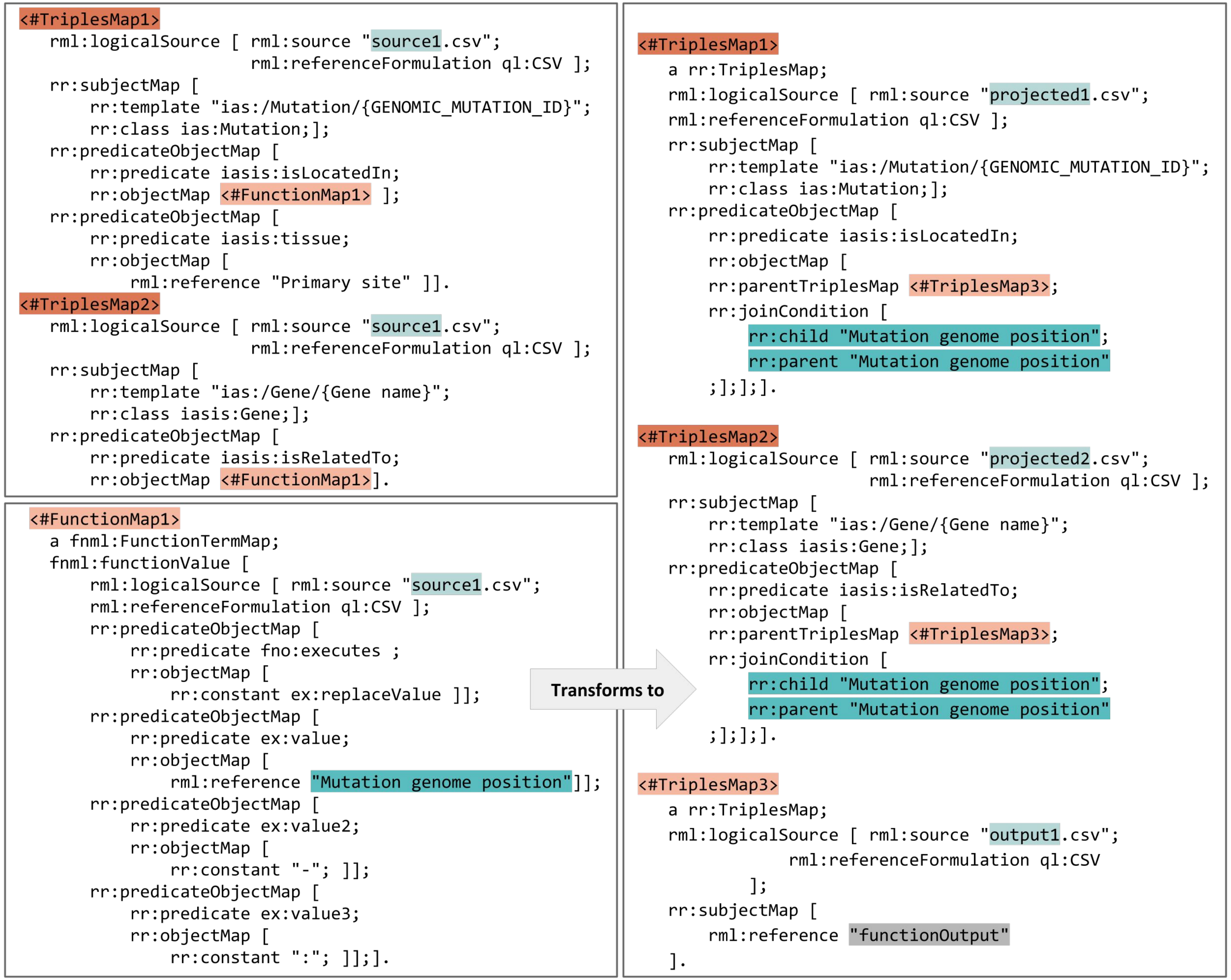
Q2) What is the impact of different types of complexity over transformation functions during a knowledge graph creation process?

Q3) How does the repetition of a same function in different mappings affect the existing RML engines?

FunMap- Architecture



FunMap - Heuristic I

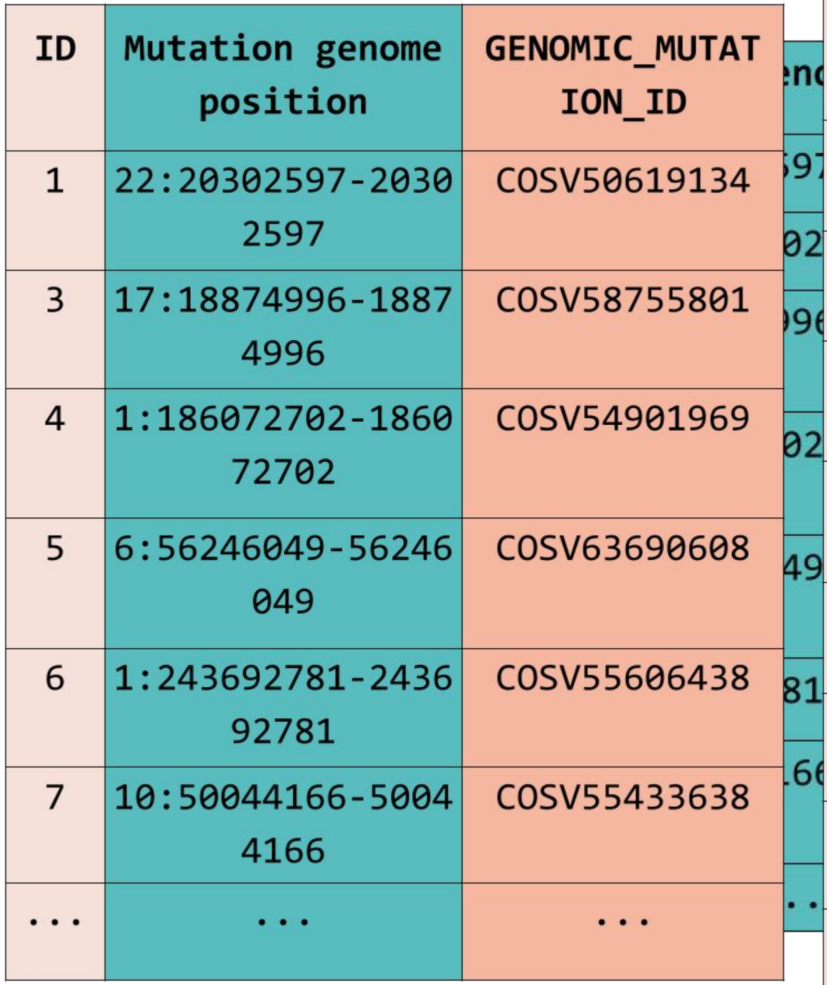


```
<#TriplesMap1>
   rml:logicalSource [ rml:source "source1.csv";
                       rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rr:template "ias:/Mutation/{GENOMIC_MUTATION_ID}";
       rr:class ias:Mutation;];
   rr:predicateObjectMap [
       rr:predicate iasis:isLocatedIn;
       rr:objectMap <#FunctionMap1> ];
   rr:predicateObjectMap [
       rr:predicate iasis:tissue;
       rr:objectMap [
           rml:reference "Primary site" ]].
<#TriplesMap2>
   rml:logicalSource [ rml:source "source1.csv";
                       rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rr:template "ias:/Gene/{Gene name}";
       rr:class iasis:Gene;];
   rr:predicateObjectMap [
       rr:predicate iasis:isRelatedTo;
       rr:objectMap <#FunctionMap1>].
```

```
 <#FunctionMap1>
   a fnml:FunctionTermMap;
   fnml:functionValue [
       rml:logicalSource [ rml:source "source1.csv";
       rml:referenceFormulation ql:CSV ];
       rr:predicateObjectMap [
           rr:predicate fno:executes ;
           rr:objectMap [
               rr:constant ex:replaceValue ]];
       rr:predicateObjectMap [
           rr:predicate ex:value;
           rr:objectMap [
               rml:reference "Mutation genome position"]];
       rr:predicateObjectMap [
           rr:predicate ex:value2;
           rr:objectMap [
               rr:constant "-"; ]];
       rr:predicateObjectMap [
           rr:predicate ex:value3;
           rr:objectMap [
               rr:constant ":"; ]];].
```

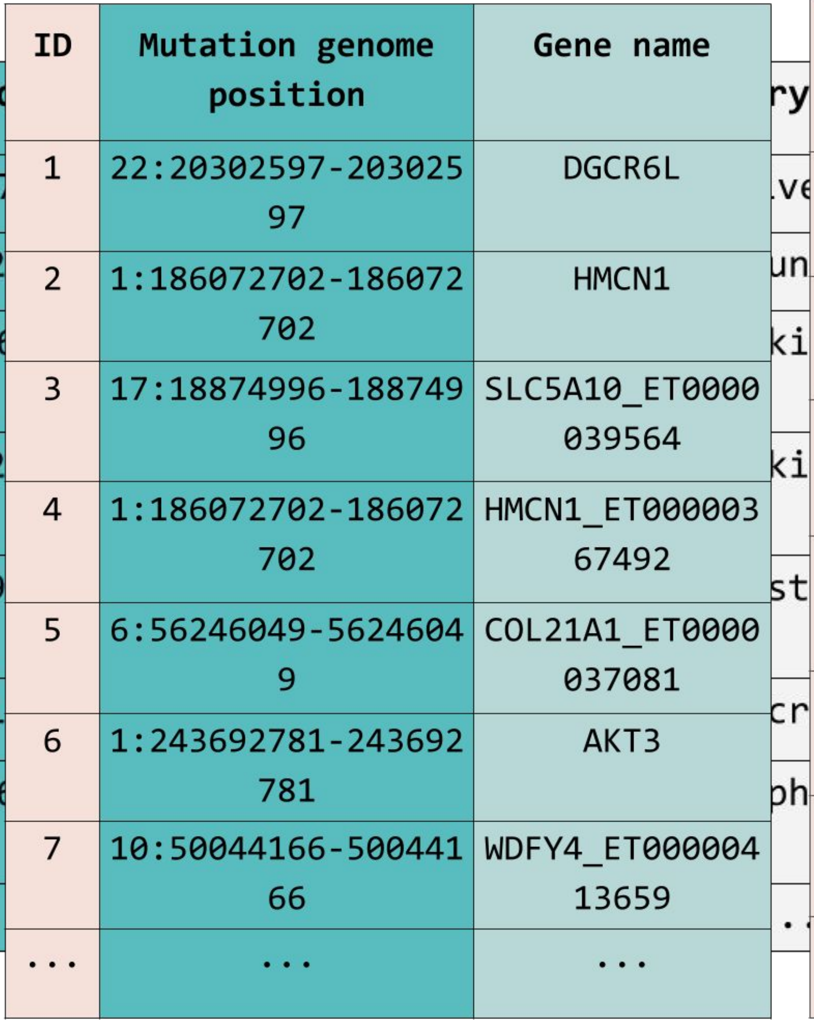
```
<#TriplesMap1>
   a rr:TriplesMap;
   rml:logicalSource [ rml:source "projected1.csv";
   rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rr:template "ias:/Mutation/{GENOMIC_MUTATION_ID}";
       rr:class ias:Mutation;];
   rr:predicateObjectMap [
       rr:predicate iasis:isLocatedIn;
       rr:objectMap [
       rr:parentTriplesMap <#TriplesMap3>;
       rr:joinCondition [
           rr:child "Mutation genome position";
           rr:parent "Mutation genome position"
       ;];];].

<#TriplesMap2>
   rml:logicalSource [ rml:source "projected2.csv";
                       rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rr:template "ias:/Gene/{Gene name}";
       rr:class iasis:Gene;];
   rr:predicateObjectMap [
       rr:predicate iasis:isRelatedTo;
       rr:objectMap [
       rr:parentTriplesMap <#TriplesMap3>;
       rr:joinCondition [
           rr:child "Mutation genome position";
           rr:parent "Mutation genome position"
       ;];];].

<#TriplesMap3>
   a rr:TriplesMap;
   rml:logicalSource [ rml:source "output1.csv";
               rml:referenceFormulation ql:CSV
         ];
   rr:subjectMap [
       rml:reference "functionOutput"
   ].
```

FunMap - Result Heuristic I



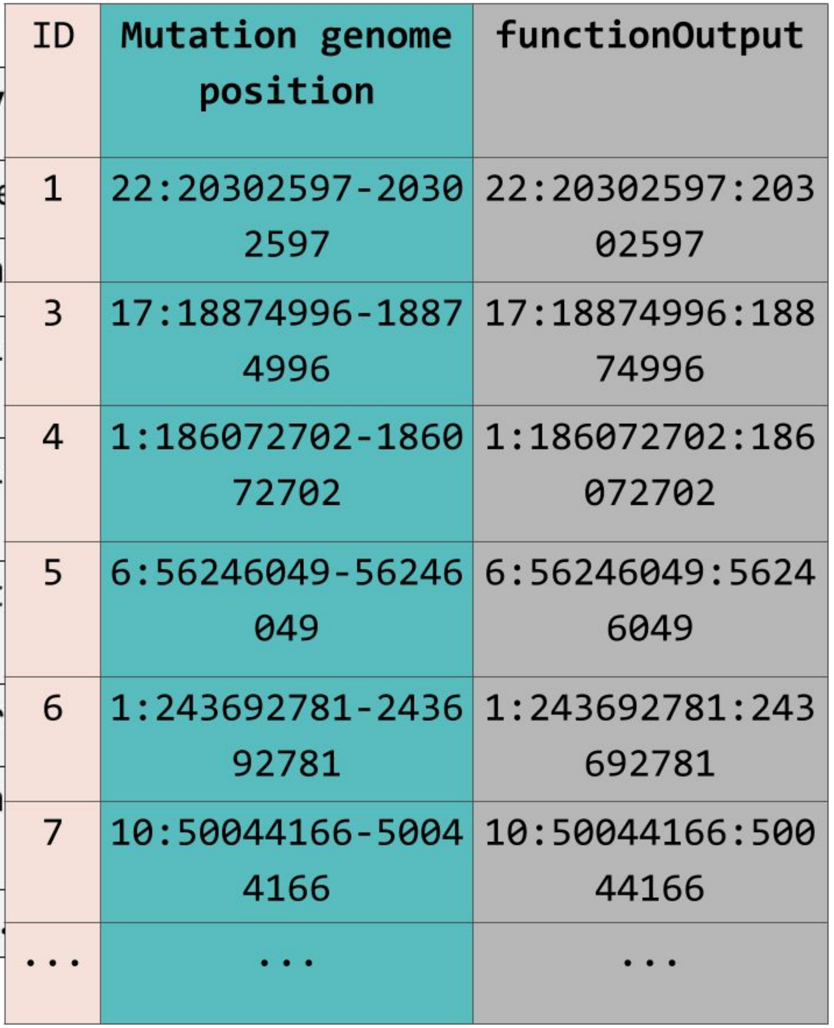
ID	Mutation genome position	GENOMIC_MUTATION_ID
1	22:20302597-20302597	COSV50619134
3	17:18874996-18874996	COSV58755801
4	1:186072702-186072702	COSV54901969
5	6:56246049-56246049	COSV63690608
6	1:243692781-243692781	COSV55606438
7	10:50044166-50044166	COSV55433638
...	...	...

Projected 1



ID	Mutation genome position	Gene name
1	22:20302597-20302597	DGCR6L
2	1:186072702-186072702	HMCN1
3	17:18874996-18874996	SLC5A10_ET0000039564
4	1:186072702-186072702	HMCN1_ET00000367492
5	6:56246049-56246049	COL21A1_ET0000037081
6	1:243692781-243692781	AKT3
7	10:50044166-50044166	WDFY4_ET00000413659
...	...	...

Projected 2



ID	Mutation genome position	functionOutput
1	22:20302597-20302597	22:20302597:20302597
3	17:18874996-18874996	17:18874996:18874996
4	1:186072702-186072702	1:186072702:186072702
5	6:56246049-56246049	6:56246049:56246049
6	1:243692781-243692781	1:243692781:243692781
7	10:50044166-50044166	10:50044166:50044166
...	...	...

Output 1

FunMap - Heuristic II

```
<#TriplesMap1>
   rml:logicalSource [ rml:source "source1.csv";
                       rml:referenceFormulation ql:CSV ];
   rr:subjectMap <#FunctionMap1> ;
   rr:predicateObjectMap [
       rr:predicate iasis:represents;
       rr:objectMap [
           rml:reference "Mutation" ]];
   rr:predicateObjectMap [
       rr:predicate iasis:tissue;
       rr:objectMap [
           rml:reference "Primary site" ]].
```

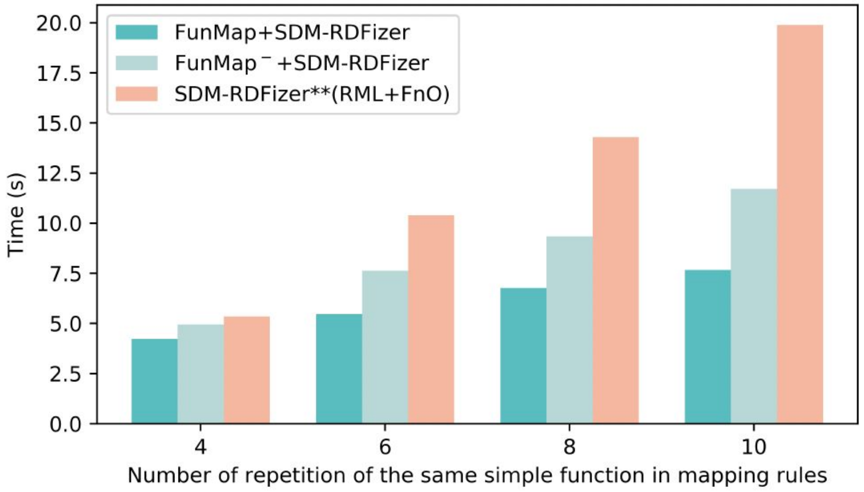
```
<#FunctionMap1>
   a fnml:FunctionTermMap;
   fnml:functionValue [
       rml:logicalSource [ rml:source "source1.csv";
       rml:referenceFormulation ql:CSV ];
       rr:predicateObjectMap [
           rr:predicate fno:executes ;
           rr:objectMap [
               rr:constant ex:replaceValue ]];
       rr:predicateObjectMap [
           rr:predicate ex:value;
           rr:objectMap [
               rml:reference "Mutation genome position"]];
       rr:predicateObjectMap [
           rr:predicate ex:value2;
           rr:objectMap [
               rr:constant "-"; ]];
       rr:predicateObjectMap [
           rr:predicate ex:value3;
           rr:objectMap [
               rr:constant ":"; ]];].
```

Transforms to

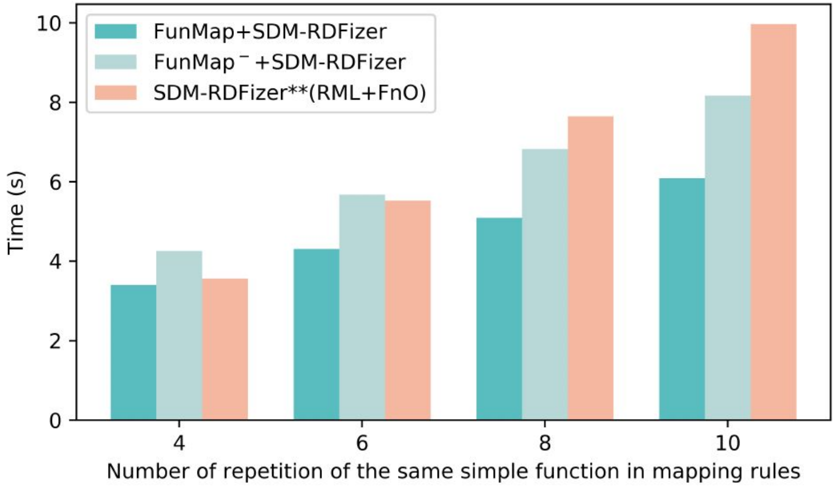
```
<#TriplesMap1>
   rml:logicalSource [ rml:source "output1.csv";
                       rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rml:reference "functionOutput" ];
   rr:predicateObjectMap [
       rr:predicate iasis:represents;
       rr:objectMap [
       rr:parentTriplesMap <#TriplesMap2> ;
       rr:joinCondition [
           rr:child "Mutation genome position";
           rr:parent "Mutation genome position";];];].
   rr:predicateObjectMap [
       rr:predicate iasis:tissue;
       rr:objectMap [
       rr:parentTriplesMap <#TriplesMap3> ;
       rr:joinCondition [
           rr:child "Mutation genome position";
           rr:parent "Mutation genome position";];];].
<#TriplesMap2>
   a rr:TriplesMap;
   rml:logicalSource [ rml:source "projected1.csv";
                   rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rml:reference "Mutation" ].
<#TriplesMap3>
   a rr:TriplesMap;
   rml:logicalSource [ rml:source "projected1.csv";
                   rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rml:reference "Primary site" ].
```

FunMap over SDM-RDFizer - Experimental results

Simple functions (lower, upper)

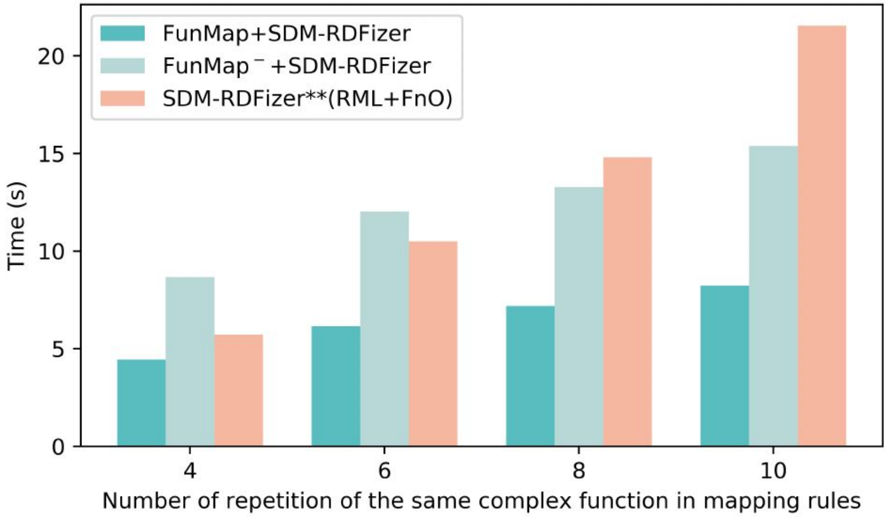


(a) SDM-RDFizer - 25% of duplicates

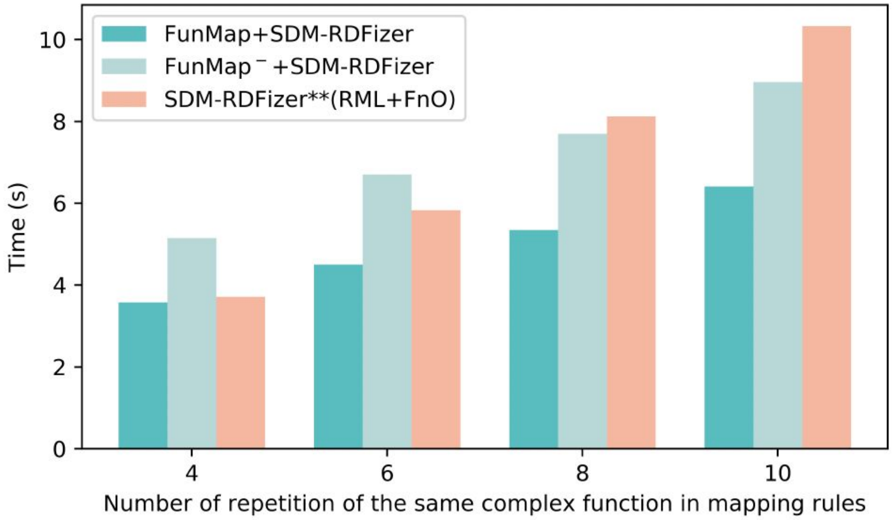


(b) SDM-RDFizer - 75% of duplicates

Complex functions (if, replace, multiple columns)



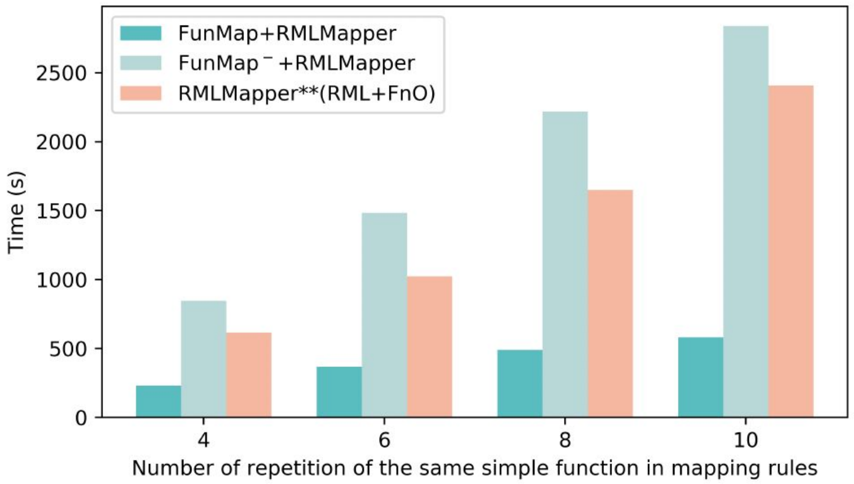
(a) SDM-RDFizer - 25% of duplicates



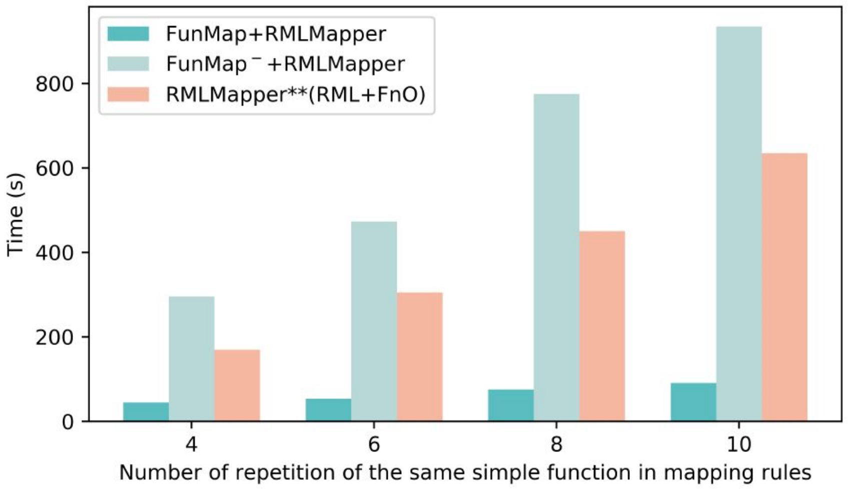
(b) SDM-RDFizer - 75% of duplicates

FunMap over RMLMapper - Experimental results

Simple functions (lower, upper)

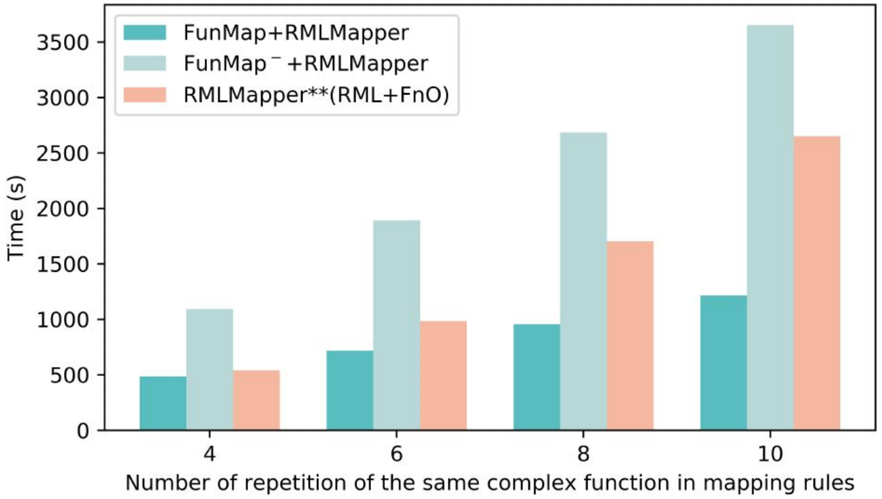


(c) RMLMapper - 25% of duplicates

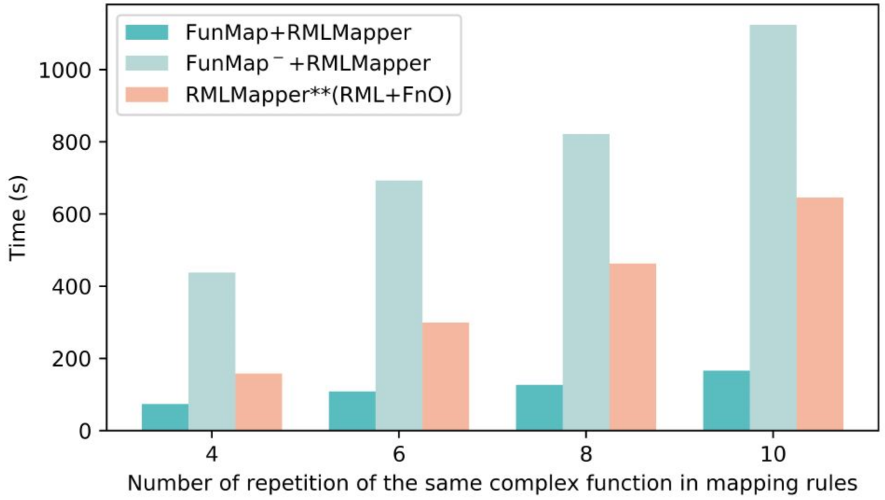


(d) RMLMapper - 75% of duplicates

Complex functions (if, replace, multiple columns)



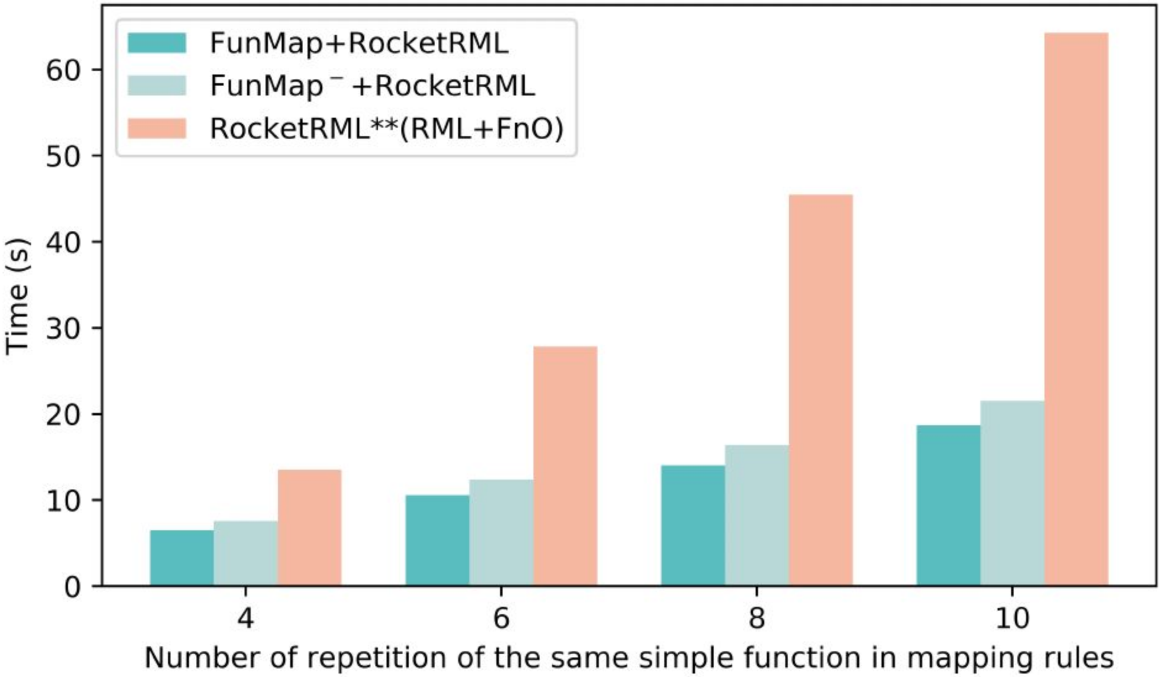
(c) RMLMapper - 25% of duplicates



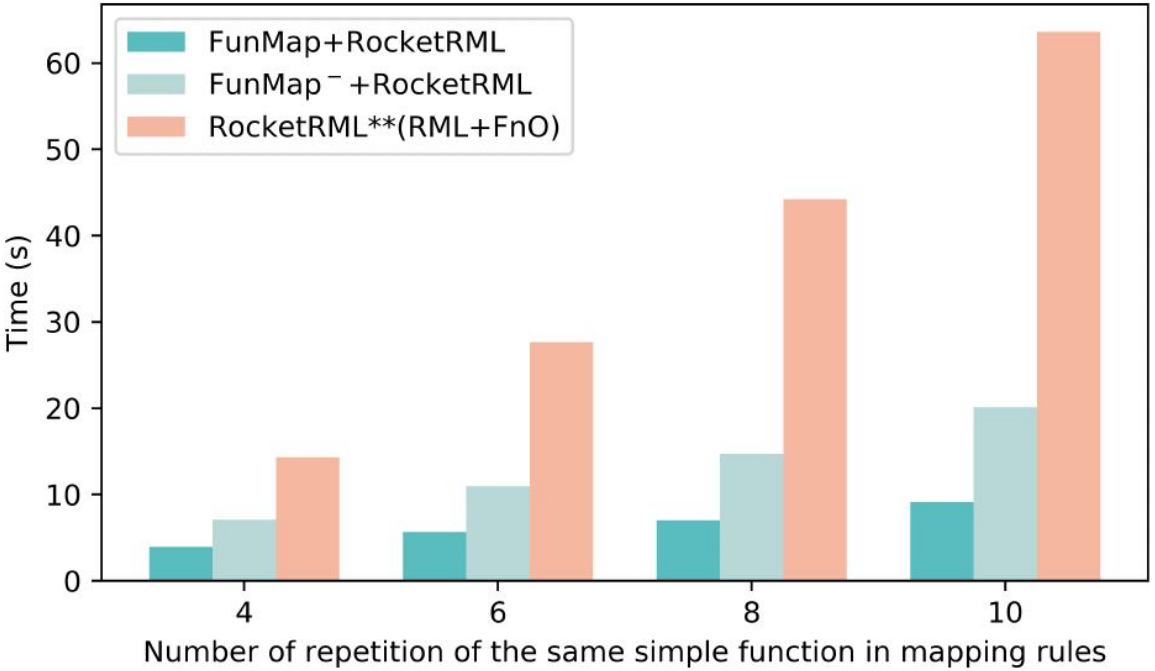
(d) RMLMapper - 75% of duplicates

FunMap over RocketRML - Experimental results

Simple functions (lower, upper)



(e) RocketRML - 25% of duplicates



(f) RocketRML - 75% of duplicates

Conclusions

- Heuristic-based approach for **generating scalable data integration systems**
- FunMap converts data integration systems in RML+FnO into **equivalent data integration systems specified in RML**
- FunMap generates data integration systems that **enhance RML-complaint engines**
- Empirical evaluations suggest that the execution time of RML+FnO **is reduced by up to 20 times**

Lessons Learned

- Research takes time:
  - SDM-RDFizer: March 2019 - June 2020
  - FunMap: October 2019 - May 2020
- (Try to) be on the same page with your co-authors/supervisors
- Be passionate and believe in what you do
- Envision big/general/global and start small/specific/local
- Be patient with (Semantic Web) reviewers
- Your impact will be as big as the quality of your pitch/paper to explain the solution*

*Pieter Colpaert at Open Summer of Code

From Engineering To Research

The KG Construction Use Case

David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain

Samaneh Jozashoori, SDM-TIB
Maria-Esther Vidal, SDM-TIB
Enrique Iglesias, University of Bonn
Diego Collarana, Fraunhofer IAIS
Oscar Corcho, OEG-UPM

✉ dchaves@fi.upm.es dchavesf

23/07/2020 Online