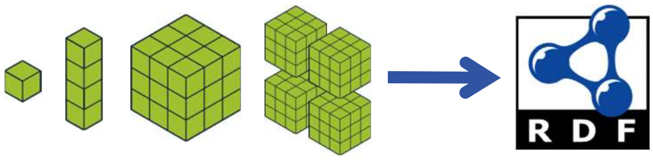
Representing and querying data tensors in RDF and SPARQL

Piotr Marciniak [1], Piotr Sowiński [1,2], Maria Ganzha [1,3]
[1] Warsaw University of Technology, [2] NeverBlink, [3] Systems Research Institute, Polish Academy of Sciences

Want to use tensors and RDF together?

- ✘ Clunky vector DB references
- ✘ Queries across multiple DBs
- ✘ Data sync issues
- ✘ No native tensor datatypes in RDF
- ✘ No support for vector ops in SPARQL

What if you could put tensors directly *in* RDF?



- ✔ Vectors & RDF graphs in one DB
- ✔ Native RDF datatypes
- ✔ Native SPARQL ops
- ✔ 1D, 2D, 3D and beyond
- ✔ Perfect for RAGs & multimodal data

Motivation

- **Data tensors are everywhere:** ML embeddings (including LLMs), images, videos, and more.
- Neurosymbolic AI systems must process **both** RDF and tensors.
- Current RDF structures are too slow, inefficient, and lack native SPARQL operators for tensors.

Method

Two new RDF datatypes for tensors:

- `dt:BooleanDataTensor`

```
PREFIX dt: <https://w3id.org/rdf-tensor/datatypes#>

:s :p
"{\"shape\":[1, 3],\"data\":[true, false, true]}"^^dt:BooleanDataTensor .
```

- `dt:NumericDataTensor`

```
:s :p1
 "{\"type\":\"int32\",\"shape\":[2],\"data\":[1, 2]}"^^dt:NumericDataTensor .
:s :p2
 "{\"type\":\"float32\",\"shape\":[2],\"data\":[1e0, 2.0]}"^^dt:NumericDataTensor
.
```

36 new SPARQL functions:

- **Elementwise transformations** (e.g., cosine – `dtf:cos`)
- **Binary operators** with broadcasting (e.g., `dtf:add`, `dtf:or`)
- **Indexing functions** for tensor slicing (e.g., `dtf:getSubDT`)
- **Concatenation functions** (e.g., `dtf:concat`)
- **Reduction functions** (e.g., `dtf:sum`, `dtf:avg`)
- **Similarity functions** (e.g., `dtf:euclideanDistance`)

```
PREFIX dtf: <https://w3id.org/rdf-tensor/functions#>
SELECT * WHERE {
    :s :p1 ?dt1 .
    :s :p2 ?dt2 .
    BIND(dtf:cosineSimilarity(?dt1, ?dt2) AS ?cos)
    BIND(dtf:norm1(0, dtf:minus(?dt1, ?dt2)) AS ?norm_dt1)
}
```

4 new SPARQL aggregates:

`dta:sum, dta:avg, dta:std, dta:var`

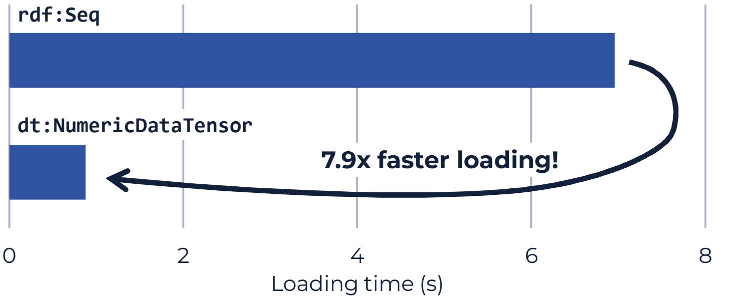
```
PREFIX dta: <https://w3id.org/rdf-tensor/aggregates#>

SELECT ?s (dta:sum(?dt) AS ?sum_tensor) (dta:avg(?dt) AS ?avg_tensor)
WHERE {
    ?s :p1|:p2 ?dt .
} GROUP BY ?s
```

Implementation and results

The 2 new datatypes, 36 SPARQL functions, and 4 SPARQL aggregates are **fully implemented as an extension for Apache Jena** and released under the Apache 2.0 license. The extension is available along with an example dataset on GitHub.

In the example dataset (scientific paper metadata + text embeddings), the file size using `dt:NumericDataTensor` was reduced by a factor of **2.1**, and the loading time decreased by a factor of **7.9** as compared to having the vectors represented in `rdf:Seq`.



Conclusion

- The proposed method for RDF tensor representation is **much more efficient** than pure-RDF approaches.
- New SPARQL functions and aggregates allow for **easy manipulation of tensors in queries**.
- This work **answers a direct industry need** for direct integration of RDF graph data with tensors in neurosymbolic AI.
- We publish the code, specification, documentation, and example dataset as **open-source assets**.
- **We are seeking to gather feedback from the community, refine the specification, and move towards industry adoption.**

Code & spec

Paper

https://w3id.org/rdf-tensor