



# Moving from Tabular Knowledge Graph Quality Assessment to RDF Triples Leveraging ChatGPT

DQMLKG'24: Data Quality meets Machine Learning and Knowledge Graphs

**Gabriele Tuozzo**

Master's degree student in computer science at the  
University of Salerno

# Table of contents



1. Introduction
2. Move from the CSV returned by KGHeartBeat to RDF triples
3. Evaluation of the result
4. Future directions
5. Conclusion

# Introduction

- The goal is to show **one of the possible approaches of using Machine Learning (ML)** with **Knowledge Graphs (KGs)**, not to involve it in quality measurement, but to **manipulate** the data from a measurement already performed.
- We want to understand the limitations of Large Language Models (LLMs) applied to the context of **generating and manipulating** KGs quality data.
- Can these technologies effectively handle this data and **relieve the user of manual tasks**?

# Introduction

- Data quality is a **multidimensional problem** encompassing heterogeneous and **multiple quality dimensions**, including but not limited to **accessibility**, **interlinking**, **performance**, **syntactic validity**, and **completeness**
- Quality dimensions are rather abstract, they can be **measured** via quality assessment **metrics** which rely on **quality indicators**.
- Suppose we are in a context where an assessment tool returns a **tabular representation of quality**, in the direction of **eat our own food**, it may be useful to have an **RDF representation of this assessment**, i.e. to build a KG on the basis of quality results

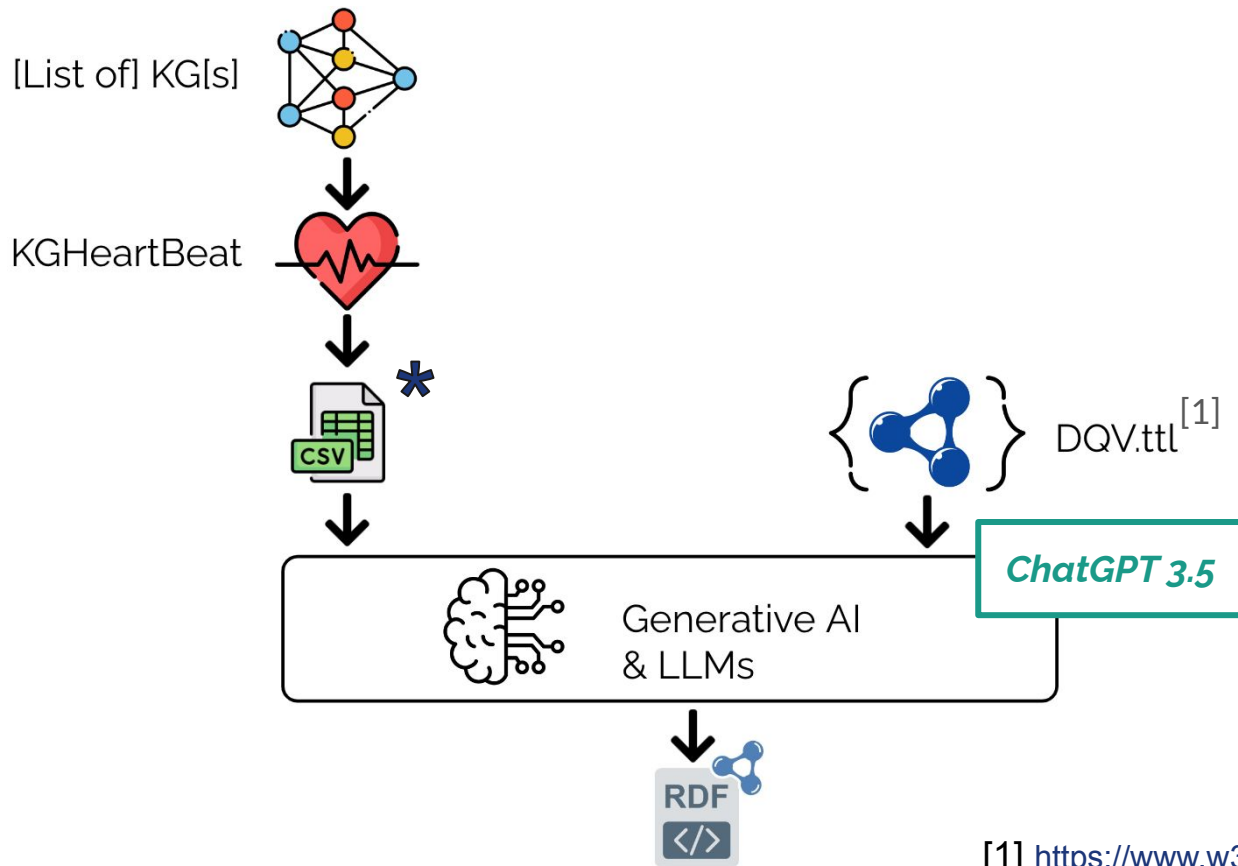
# Introduction - KGHeartBeat

- The quality data about the KGs used in this work are calculated by **KGHeartBeat [1]**, which will be presented as a **demo at ESWC 2024**
- KGHeartBeat is a **fully-automatic** community-shared open source quality assessment tool to **periodically** perform quality analysis on all the **freely available** KGs
- The KGHeartBeat **web-application** can be configured to query a list of KGs and implements a large set of KG quality metrics proposed by **Zaveri et al. [2]** belonging to **different quality dimensions**

[1] <https://github.com/isislab-unisa/KGHeartbeat>

[2] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, S. Auer, Quality assessment for linked data: A survey, Semantic Web 7 (2016) 63–93. doi: 10.3233/SW- 150175.

# Move from the CSV to RDF triples



\* Revised version

[1] <https://www.w3.org/TR/vocab-dqv/>

## Move from the CSV to RDF triples - Scenarios considered

- ***Single dimension, multiple metrics:*** we focus only on **availability** dimension
- ***Single dimension category, multiple dimensions and multiple metrics:*** we focus on the **trust category** of ten different KGs
- ***Different dimensions categories:*** we consider a CSV of ten different KGs randomly selected with a set of **quality dimensions related to different categories**



### 3. Evaluation of the result

We will briefly report **the prompts to chatGPT** with some considerations on the answer.



# Single dimension, Multiple Metrics: the Availability Case



1. Consider the following csv entitled "availability\_scores.csv":<PASTE CSV FILE CONTENT>.
2. Consider the following ontology in ttl format entitled "dqv.ttl": <PASTE ONTOLOGY IN TTL FORMAT>.
3. Can you model the "availability\_scores.csv" file content according to the "dqv.ttl" ontology and return the resulting triples in rdf format?

```
:d1 a dqv:QualityMeasurementDataset;  
  rdfs:label "(CZ-NACE)" .  
:dqv:metric1 a dqv:Metric;  
  rdfs:label "Availability_sparqlEndpoint" .  
:d1_measure a dqv:QualityMeasurement;  
  dqv:isMeasurementOf :dqv:metric1;  
  dqv:value "Available" .
```

Snippet



# Considerations

---




- Correctly **attaches** each value to a **modeled metric**
- Correctly creates the correspondence between **metric scores** and **datasets**



- Only returns a **skeleton** of the RDF file

# Single Dimension Category, Multiple Dimensions and Multiple Metrics: the Trust Case

- 
1. Consider the following csv entitled "trust\_scores.csv": <PASTE CSV FILE CONTENT>.
  2. Consider the following ontology in ttl format entitled "dqv.ttl": <PASTE ONTOLOGY IN TTL FORMAT>
  3. Let's consider that the CSV file contains **all dimensions concerning the trust category** and for each dimension, the file details its metrics. To distinguish metrics and dimension, consider that all the file column names **follow the pattern of DIMENSION\_METRIC**. With these premises, can you model the data contained in csv file according to the "dqv.ttl" ontology and return the **complete and detailed set of resulting triples in rdf format?**

# Considerations



- **Correctly** recognizes **category**, **dimensions** and **metrics** and models them **as an hierarchy**.



- It **stops** to a **single metric** and to a **single KG**.
- Even if we suggest to **focus on a single KG**, it returns an incomplete formulation, **just modeling a single metric** and suggesting to add similar triples for other metrics and dimensions.


And if we ask to focus on a single dimension (in the Trust category)?

# Considerations



- Even when **we ask to focus on a single dimension**, e.g., the **believability dimension**, and a **single KG**, it returns a **incomplete formulation**, completed by explicitly asking for it with three iterative interaction

# Different dimensions categories cases

- 
1. Consider the following csv entitled "trust\_scores.csv": <PASTE CSV FILE CONTENT>.
  2. Consider the following ontology in ttl format entitled "dqv.ttl": <PASTE ONTOLOGY IN TTL FORMAT>
  3. Considering that the csv file pasted before contains scores attached to different dimensions and metrics. To distinguish metrics and dimension, consider that all the file column names follow the pattern of DIMENSION\_METRIC. All the column names ending with ScoreValue represent the score attached to the dimension reported as prefix of the column name. With these premises, can you model the data contained in csv file related to the **KG entitled "DBpedia in French"** according to the "dqv.ttl" ontology and return **the complete and detailed set of resulting triples in rdf format** both reporting the score of all the dimensions and detailing all their metrics' measurements?

# Considerations

---



- Return a skeleton **correctly modeling** the **dimension category**, **dimensions** and **metrics hierarchy** but ...



- ... **sends the user** the role of instantiating the **names of the metrics and dimensions** and **completing the triples** by replicating the identified patterns

# Conclusion




- ChatGPT returns a useful skeleton for modeling the CSV content (once **the structure of the csv file has been explained**)
- Instead, automatically understands the structure of the **ontology without any clarification.**
- But the more data there is in the csv file, the more the skeleton **only reports the structure that must be manually replicated.**
- **The trick** used to complete the RDF file work only when the focus is on a **single dimension** and a **single KG**
- ChatGPT successfully saved human effort but **requires human expertise** in **assessing** and **refining** its outcome



# Future directions



- We want to consider the effect of prompting ChatGPT with just a **portion of the DQV** and compare the results
- Investigate the **utility** of a **semi-automatic** KG generation
- Compare the KG quality returned by ChatGPT with the one obtained with a **manual traditional approach**



# Thank you for your attention!

Any questions?

All the material seen during this presentation (**the full iterations with chatGPT** and the **CSV files used**) are available and freely accessible on GitHub: <https://bit.ly/KGHB-Workshop>

Contacts

Email: [gtuozzo@unisa.it](mailto:gtuozzo@unisa.it)

Linkedin: <https://bit.ly/gtuozzo>