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

Gabriele Tuozzo[1]

[1]Dipartimento di Informatica, Unviersità degli Studi di Salerno, Fisciano (SA), ITALY

Abstract

Data quality assessment is a multifaceted challenge involving various dimensions such as accessibility, interlinking, and completeness. These dimensions are domain-dependent and can be aggregated into a score between 0 and 1, facilitating dataset ranking based on quality. Achieving effective representation and explanation of these rankings poses significant challenges akin to those in machine learning, where interpretability and understandability are crucial. In the domain of natural language processing, data interpretation is a critical yet complex process, often requiring domain expertise and significant resources. Advanced Language Model Models (LLMs) offer promise in automating annotation tasks, ensuring consistency, and adapting to specific domains. Leveraging such models for knowledge representation tasks necessitates adept prompt engineering. This study focuses on experiencing state-of-the-art prompt engineering methods, particularly using GPT-3.5, for representing knowledge related to dataset quality. By exploring techniques to extract RDF triples from textual data without predefined labels or constraints, this work aims to enhance interpretability and understanding of dataset quality assessment results while verifying the feasibility on automatic knowledge representation leveraging LLMs.

Keywords

Quality assessment, Knowledge extraction, Interpretability, Prompt engineering, GPT-3.5

1. 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 [1]. The significance of each dimension depends on the domain or particular use cases. The result returned by assessing these quality dimensions can be combined to generate a score ranging from 0 to 1, allowing datasets to be ranked accordingly, where a higher score indicates a higher quality.

Once datasets are ranked based on their quality score, the key challenges are how to effectively represent the quality of the datasets and how to explain the produced results. Analogous to Machine Learning, explainability is often replaced with the notion of interpretability [2], which are considered interchangeable terms within the broader Artificial Intelligence (AI) community and particularly among scholars specializing in automated learning and reasoning. Conversely, the software engineering community prefers the term understandability [3]. In general terms, interpretability is often defined as the ability to convey or extract the meaning of an abstract

DQMLKG'24: Data Quality meets Machine Learning and Knowledge Graphs, DQMLKG Workshop at ESWC 2024, May 26th or 27th, 2024, Hersonissos, Greece

g.tuozzo4@studenti.unisa.it (G. Tuozzo)

0000-0002-9421-8566 (G. Tuozzo)

concept, whereas understandability refers to the capability of making it comprehensible to end-users [2]. This work focuses on the interpretability and explicability of quality scores, treating those terms as synonyms.

Knowledge Graphs (KGs) are often preferred in scenarios where explainability and interpretability are crucial [4], as they explicitly represent relationships between entities and provide a structured knowledge representation. Hence, a crucial step in the direction of interpreting quality scores is modeling them as a KG. The process of KG construction requires a significant amount of manual effort and expert knowledge in identifying and labeling sentences or patterns, performing named entity recognition, relation extraction, and semantic parsing [5]. Often, this process encompasses at least the involvement of experts in the modeled field and computer scientists or experts in the targeted ontology. As a matter of fact, in the complex realm of machine learning and natural language processing (NLP), data annotation stands out as a critical yet challenging step, transcending simple label attachment to encompass a rich array of auxiliary predictive information. Data annotation poses significant challenges for current machine learning models due to the complexity, subjectivity, and diversity of data, requiring domain expertise and the resource-intensive nature of manually labeling large datasets.

Cutting-edge Large Language Models (LLMs) such as GPT-3.5[1], Gemini[2] and Llama-2[3] offer a promising opportunity to revolutionize data annotation. LLMs play a pivotal role in enhancing the accuracy and efficiency of data annotation processes. Their ability to automate annotation tasks, ensure consistency across large volumes of data, and adapt through fine-tuning or prompting for specific domains, mitigates challenges inherent in traditional methods, setting a new standard for NLP [6]. By providing a well-devised input sequence (e.g. a textual prompt), LLMs can adeptly undertake knowledge extraction tasks. However, the question is how to formulate an effective input prompt, and this is where prompt engineering assumes significance.

In our investigation, we consider the free GPT-3.5 a state-of-the-art model for prompt engineering to explore prompt engineering's role in knowledge extraction and representation of tabular datasets quality. We adapt to different situations that generate knowledge in the context of knowledge extraction [6]. Specifically, modeling quality dimensions and metrics score as a CSV file, we aim to extract all potential triples from the text, without supplying any predetermined labels nor imposing constraints within the prompt as in Closed Information Extraction settings. This work provides the adaptation of state-of-the-art prompt engineering methods in the context of knowledge extraction from the quality of the datasets. It introduces a first step through the interpretability of the final score, verifying the feasibility of an automatic process of knowledge extraction leveraging LLMs.

This paper is organized as follows: Section 2 provides the background on quality and discusses the related work on knowledge extraction and representation. Section 3 documents the performed approach reported in such detail to enable reproducibility. Section 4 introduces the preliminary results of our approach, which are discussed in Section 5. We report conclusive thoughts in Section 6 along with future directions.

[1]OpenAI GPT-3.5: https://openai.com/gpt-4
[2]Google Gemini: https://gemini.google.com
[3]Meta Llama-2: https://llama.meta.com

2. Background

2.1. Terminology

Data quality assessment involves the measurement of quality dimensions relevant to the consumer and considering dataset characteristics, which can be grouped in dimension clusters. Inspired by Zaveri et al. [1], we consider the following dimension clusters:

- *Accessibility dimensions* which involves aspects related to the access, authenticity, and retrieval of data to obtain either the entire or some portion of the data (or from another linked dataset) for a particular use case. It includes availability, licensing, security, and performance.
- *Intrinsic dimensions* are those that are independent of the user's context. It includes semantic accuracy, consistency, and conciseness. These dimensions focus on whether the information (syntactically and semantically) correctly and compactly represents the real world and whether the information is logically consistent in itself.
- *Contextual dimensions* are those that highly depend on the context of the task at hand, assessing the amount of published data, their relevancy, trustworthiness, understandability, and timeliness. This dimension can be further refined by considering the following aspects as separate dimension clusters:
  - *Trust dimensions* focusing on trustworthiness in terms of verifiability, reputation, and believability;
  - *Dataset dynamicity* focusing on the currency and timeliness.
- *Representational dimensions* capture aspects related to the design of the data, such as representational conciseness, interoperability, interpretability, and versatility.

While quality dimensions are rather abstract, they can be measured via quality assessment metrics which rely on quality indicators. An assessment score is computed from these indicators using a scoring function.

2.2. Related work

Extracting knowledge from tabular data like databases, Web tables and CSV files is a common way for KG construction. If users are aware of tables semantics, they can define and use heuristic rules to transform their data into KG facts. However, usually end-users lack a deep understanding of tables meta information, such as table name and column header. In recent years, transformer-based LMs have been investigated for processing tables and representing learning, such as TURL [7], RPT [8], Starmie [9]. There have been several attempts that use LLMs for these tasks, such as Doduo [10] which focuses on the prediction of column types and identification of inter-column relationships, and Korini et al. [11] who prompt ChatGPT to annotate semantic column types. Some attention has been given also to utilizing LLMs for tabular data processing and KG construction, such as the work authored by Kommineni et al. [12]. However, there is still room for investigation, mainly in representing non-textual tabular data, like numbers [13]. Moreover, LLMs are mostly applied to process and understand

tables but rarely applied to the final step of knowledge extraction [13]. As some examples in this direction, OntoGPT [14] and Trajanoska et al. [15] extract instances from texts to populate an ontology, but there are no counterparts for tables. Our contribution target this direction, exploring how to populate an ontology using ChatGPT starting from a tabular representation of quality scores. It goes in the direction of leveraging LLMs to represent and interpret KG quality results integrating LLMs in the KG quality measurement pipeline as a way to enhance the interpretation of quality assessment report. As a result, using LLMs to automatically convert CSV data quality assessment to RDF triples, make assessment results machine-readable and potentially useful for an automatic elaboration.

3. LLM-driven Knowledge Retrieval Process

This section describes the performed process to move from CSV quality assessment returned by KGHeartBeat[4] to RDF triples leveraging ChatGPT.

KGHeartBeat is a fully-automatic community-shared open source quality assessment tool to periodically perform quality analysis on all the freely available KGs that can be automatically retrieved by widely used data and knowledge aggregation platforms, such as LOD Cloud[5] and DataHub[6]. 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. [1] belonging to different quality dimensions, focusing on those that can be automatically and objectively computed without requiring a gold standard. The implementation details of all the supported quality dimensions and the related metrics are freely accessible online[7]. Once selected the quality dimensions of interest, quality results can be downloaded as CSV files. The CSV stores a KG quality assessment per line, listing all the metrics' and dimensions' scores as columns. As a convention, all the metrics related to the same dimensions share as prefix the name of the dimension. Per each dimension, there is both a weighted and a normalized score. Such as an example, the CSV file focused on the availability dimension results concerning a single KG is structured as follows reporting the header in bold and the value attached to it:

- **kg_id** - cznace
- **analysis_date** - 2024-01-28
- **Availability_sparqlEndpoint** - Available
- **Availability_RDFDumpM** - -1
- **Availability_RDFDumpQ** - True
- **Availability_inactiveLinks** - True

The quality assessment CSV files can be used along with an ontology to model KG quality scores as input to LLM, such as ChatGPT, as summarized in the process visible in Figure 1. We

[4]KGHeartBeat: http://www.isislab.it:12280/kgheartbeat
KGHeartBeat GitHub repository: https://github.com/isislab-unisa/KGHeartbeat
[5]LOD Cloud: https://lod-cloud.net
[6]DataHub: https://datahub.io
[7]Metric details: https://isislab-unisa.github.io/KGHeartbeat

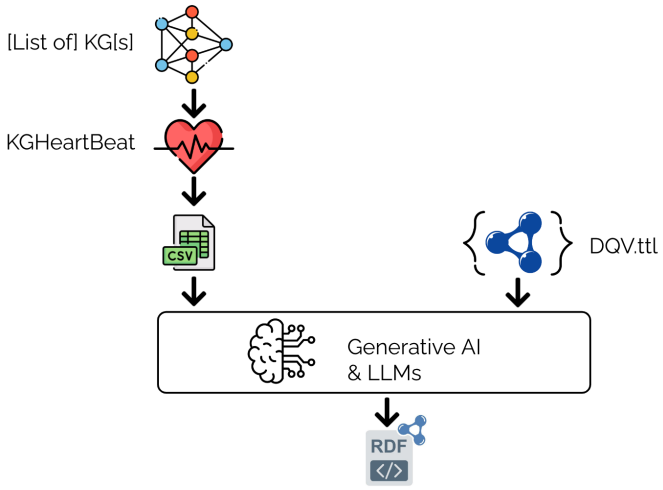

Figure 1: LLM-driven Knowledge Retrieval Process. From a list of user-selected KGs, KGHeartBeat is a community-shared software that performs quality assessment and returns dimensions and metrics scores as a CSV file. Providing LLM with a textual representation of the CSV and the TTL of the ontology used as a reference to model quality scores according to the Semantic Web technologies, we aim to obtain a triple representation of data in the CSV.

used the CSV returned by KGHeartBeat and the Data Quality Vocabulary[8](DQV) in TTL format to perform the experiments reported in this article. DQV defines quality measures as specific instances of *dqv:QualityMeasurements* and it relies on quality dimensions (*dqv:Dimension*) e.g. the availability of a dataset, and quality metrics (*dqv:Metric*) e.g. whether or not a SPARQL endpoint is accessible.

We consider three scenarios of incremental complexity, defined as follows:

- *Single dimension, multiple metrics.* The end-user focused on a single dimension. Hence, KGHeartBeat returns a CSV containing the detail for each metric implemented for the user-selected dimension and an overall score at dimension level. As an example, we focus on scores concerning *availability* of ten different randomly selected KGs.
- *Single dimension category, multiple dimensions and multiple metrics.* The end-user focused on a single dimension category, which include multiple dimensions, each computed in terms of multiple metrics. Hence, KGHeartBeat returns a CSV containing the detail for each metric implemented for the user-selected dimension and an overall score at dimension level. As an example, we focus on scores concerning the *trust* category of ten different KGs randomly selected.
- *Different dimensions categories.* According to the task or scenario of interest, end-users can select different dimensions and the related metrics. As a result, we consider a CSV of ten different KGs randomly selected with a set of quality dimensions related to different categories.

[8]Data Quality Vocabulary: https://github.com/w3c/dwbp/blob/gh-pages/dqv.ttl

4. Results

This section reports the performed experiments describing the used prompts and discussing ChatGPT replies, mainly focusing on positive aspects and what is missing. Files used as input, the transcript of conversations performed on ChatGPT 3.5, and the resulting RDF skeleton are reported in the KGHeartBeat GitHub[9], this is to promote the reproducibility of the experiment performed.

Single Dimension, Multiple Metrics: the Availability Case. The simplest interaction with ChatGPT is uploading both the CSV and the DQV.ttl ontology without any explanation, and inspect the resulting RDF triples. Consequently, we report the used prompts in the following:

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

As a result, ChatGPT correctly models each KGs as a `dqv:QualityMeasurementDataset`, correctly attaches each value to a modeled metric, and correctly creates the correspondence between metric scores and datasets. However, it considers all the columns in the CSV as metrics, without distinguishing metrics' scores to the overall score at the dimension level. Of course, we need to clarify how the CSV file must be interpreted. Moreover, ChatGPT only returns a skeleton of the RDF file, suggesting to repeat similar patterns for other KGs and quality metrics. By reformulating the third prompt as follows, we obtained a complete modeling of RDF triples of a specific KG modeled in the CSV, e.g., DBpedia in French.

```
Considering that all the columns listed in the "availability_score.csv" file having the name
    starting with "availability_" are metrics of the availability dimension which is one of the
    accessibility dimensions, the column entitle "availabilityScoreValue" is the value
    corresponding to the availability dimension, and you can ignore the other columns ending with
    score, can you model the data contained in "availability_scores.csv" file and 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 the
    availability dimension and detailing all the metrics' measurements?
```

The explanation of the CSV file is needed to correctly distinguishing quality dimensions and metrics in the ontology. The suggestion to focus on a single KG is due to the trend of ChatGPT of proposing the skeleton of the RDF triples for a small set of metrics and KGs and suggesting to complete it following similar patterns, as observed before. Moreover, if end-users do not specify the desired level of details in terms of metrics' measurements, ChatGPT only models the quality dimension score.

Single Dimension Category, Multiple Dimensions and Multiple Metrics: the Trust Case. According to the interaction described before, the used prompts specify the CSV, the ontology and describe how to interpret the CSV header as follows:

```
Consider the following csv entitled "trust_scores.csv": <PASTE CSV FILE CONTENT>.
Consider the following ontology in ttl format entitled "dqv.ttl": <PASTE ONTOLOGY IN TTL FORMAT>
```

[9]GitHub folder with examples: https://bit.ly/kgheartbeat-eswc-ws

```
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?
```

As a result, ChatGPT correctly recognizes category, dimensions and metrics and models them as an hierarchy. However, it stops to a single metric and to a single KG. Even if we suggest to focus on a single KG, such as DBpedia in French, it returns an incomplete formulation, just modeling a single metric and suggesting to add similar triples for other metrics and dimensions. Moreover, 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 to for it with three iterative interactions. The used prompt is as follows:

```
Let's focus on the Believability dimension. Can you provide with a complete rdf concerning the KG entitled "DBpedia in French" modeling it 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?
<PARTIAL RDF ENDING WITH #...(similar triples for other metrics under the Believability dimension)>
Please, complete it
<PARTIAL RDF ENDING WITH #...(similar triples for other metrics under the Believability dimension)>
Please, complete it
<PARTIAL RDF>
```

Different dimensions categories cases. Similarly to the trust case, once provided ChatGPT with the quality scores CSV, the DQV ontology and the explanation of the CSV header to distinguish metrics and dimensions, ChatGPT returns 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.

5. Discussion

This section summarizes potentialities and limitations of a LLM-driven KG generation to convert CSV into RDF according to a data quality ontology.

Automatic RDF skeleton to be manually completed. ChatGPT returns a useful skeleton for modeling the CSV content according to the data quality vocabulary requiring a minimum explanation of the CSV header to distinguish metrics' and dimensions' scores and automatically inferring the ontology structure, without requiring any clarification. However, the more data available in terms of metrics and KGs, the more the skeleton only reports the structure that must be manually replicated for each metric and KG observed. Tricks used to complete RDF files seems to work only when end-users narrow down to a focused scenario, e.g., a single dimension and a single KG. As a result, ChatGPT successfully saved human effort in proposing the initial draft of KG, but requires human expertise in assessing and refining, if needed, its outcome, as confirmed by similar work in this field [12, 14].

Light-versions of files used as input might be required. With respect to other available and freely usable LLM, ChatGPT accepts prompts long enough to paste the entire Data Quality Vocabulary at once. However, any LLM has its own prompt limit [16]. In fact, we encountered some limitations with CSV starting from the second case due to the presence of several columns populated with a list of values, such as the list of used vocabularies. To mitigate the problem, we replaced the original versions of CSV returned by KGHeartBeat with a light version replacing lists with their size. Both the original and the light version of CSV used as prompts are in the KGHeartBeat GitHub repository.

Bias in prompt-based retrieval. Our experience with susceptibility of results according to the used prompt is confirmed by the survey authored by Pan et al. [13] concerning LLMs and KGs, which report that prompt-based retrieval is biased towards prompt structure. Thus, prompt engineering is a crucial part of knowledge retrieval from LLMs.

Low hallucination. A common problem with LLMs is hallucination of results. However, we experienced a low level of hallucination. It might be justified by the explicit requirement to extract what was found in the pasted CSV within each prompt, similarly to what has been experienced by Caufield et al. [14].

6. Conclusion and Future Directions

This article explores the opportunity to automatically generate RDF triples via LLM. We initialized ChatGPT 3.5 with a (light version of) CSV modeling KGs' quality scores automatically generated by KGHeartBeat and the data quality vocabulary and we explored the returned RDF triples. ChatGPT automatically returns a skeleton of the RDF triples inferring patterns to model metrics and KG assessment that must be manually completed by end-users. According to the level of details in describing metrics, ChatGPT correctly model data quality categories, dimensions and metrics. While manual effort is mitigated, experts in the field are still required to assess the quality of the automatically extracted triples and complete them if needed.

Future directions. This article proposes a feasibility study to generate RDF triples from a CSV file modeling KG quality assessment. As a future direction we are interested to consider the effect of prompting ChatGPT with just a portion of the DQV and compare the results in terms of quality of the returned output. Moreover, we are interested in investigating the perceived utility of a (semi-)automatic KG generation as the one described in this article by inquiring experts in the field. Finally, besides qualitative insights as the ones reported in this article, it is crucial to quantify the quality of the returned output and compare the quality of the KGs resulting by ChatGPT with the one obtained with a manual traditional approach.

References

[1] 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.

[2] G. Vilone, L. Longo, Notions of explainability and evaluation approaches for explainable artificial intelligence, Information Fusion 76 (2021) 89–106.
[3] J. M. Alonso, C. Castiello, C. Mencar, A bibliometric analysis of the explainable artificial intelligence research field, in: International conference on information processing and management of uncertainty in knowledge-based systems, Springer, 2018, pp. 3–15.
[4] J. Chen, F. Lécué, J. Z. Pan, I. Horrocks, H. Chen, Knowledge-based transfer learning explanation, in: Sixteenth International Conference on Principles of Knowledge Representation and Reasoning, 2018.
[5] D. N. Nicholson, C. S. Greene, Constructing knowledge graphs and their biomedical applications, Computational and structural biotechnology journal 18 (2020) 1414–1428.
[6] Z. Tan, A. Beigi, S. Wang, R. Guo, A. Bhattacharjee, B. Jiang, M. Karami, J. Li, L. Cheng, H. Liu, Large language models for data annotation: A survey, 2024. arXiv:2402.13446.
[7] X. Deng, H. Sun, A. Lees, Y. Wu, C. Yu, Turl: Table understanding through representation learning, ACM SIGMOD Record 51 (2022) 33–40.
[8] N. Tang, J. Fan, F. Li, J. Tu, X. Du, G. Li, S. Madden, M. Ouzzani, Rpt: relational pre-trained transformer is almost all you need towards democratizing data preparation, Proceedings of the VLDB Endowment 14 (2021) 1254–1261. doi:10.14778/3457390.3457391.
[9] G. Fan, J. Wang, Y. Li, D. Zhang, R. J. Miller, Semantics-aware dataset discovery from data lakes with contextualized column-based representation learning, Proceedings of the VLDB Endowment 16 (2023) 1726–1739. doi:10.14778/3587136.3587146.
[10] Y. Suhara, J. Li, Y. Li, D. Zhang, Ç. Demiralp, C. Chen, W.-C. Tan, Annotating columns with pre-trained language models, in: Proceedings of the 2022 International Conference on Management of Data, 2022, pp. 1493–1503.
[11] K. Korini, C. Bizer, Column type annotation using chatgpt, arXiv preprint arXiv:2306.00745 (2023).
[12] V. K. Kommineni, B. König-Ries, S. Samuel, From human experts to machines: An llm supported approach to ontology and knowledge graph construction, 2024. arXiv:2403.08345.
[13] J. Z. Pan, S. Razniewski, J.-C. Kalo, S. Singhania, J. Chen, S. Dietze, H. Jabeen, J. Omeliyanenko, W. Zhang, M. Lissandrini, R. Biswas, G. de Melo, A. Bonifati, E. Vakaj, M. Dragoni, D. Graux, Large Language Models and Knowledge Graphs: Opportunities and Challenges, Transactions on Graph Data and Knowledge 1 (2023) 2:1–2:38. doi:10.4230/TGDK.1.1.2.
[14] J. H. Caufield, H. Hegde, V. Emonet, N. L. Harris, M. P. Joachimiak, N. Matentzoglu, H. Kim, S. A. Moxon, J. T. Reese, M. A. Haendel, et al., Structured prompt interrogation and recursive extraction of semantics (spires): A method for populating knowledge bases using zero-shot learning, arXiv preprint arXiv:2304.02711 (2023).
[15] M. Trajanoska, R. Stojanov, D. Trajanov, Enhancing knowledge graph construction using large language models, 2023. arXiv:2305.04676.
[16] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, N. Zhang, Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities, arXiv preprint arXiv:2305.13168 (2023).