

An open-source frontend for RDF-based knowledge graphs

Tobias Malsheimer, Hochschule der Medien Stuttgart <malsheimer@hdm-stuttgart.de>
Magnus Pfeffer, Hochschule der Medien Stuttgart <pfeffer@hdm-stuttgart.de>

Abstract

The ongoing research project "Japanese Visual Media Graph" strives to create a knowledge graph for the domain of Japanese visual media, especially manga, anime and computer games, by cooperating with fan communities on the web to re-use their curated data [1,2]. Data is collected from multiple sources and converted into the RDF format. This allows for an iterative approach to integrating and merging the information into the final knowledge graph, while preserving provenance information [3]. The resulting graph can be queried using SPARQL, but is also published as Linked Open Data using the domain mediagraph.link.

Most RDF triple stores, especially commercial solutions, come with a simple web frontend. As an open-source alternative, there is the "Pubby" frontend, which was developed in the D2ME side-project of the Europeana initiative [4]. This software is quite versatile and functional, but it is developed as a Java Web Application and needs the corresponding infrastructure to be run. Other examples would be LodView[5], another Java web application or Cliopatra[6], which uses Prolog.

The current JVMG graph has many nodes describing abstract concepts, for example works of media, visual characters or content descriptors that have outgoing links to each instance of the concept. These can in some cases exceed 100.000 links, and the corresponding web pages are extremely slow or time out in all the software solutions we tested. Thus, a new web frontend was developed as part of the project, using the Python programming language and the Django web application framework. Unlike the Java software stack, the lightweight framework allowed for rapid prototyping and easy debugging.

In order to have fast label lookups, our approach creates a single SPARQL query that retrieves all relevant data for a given URI and the corresponding labels. This minimizes the amount of connections to the database and allows the database to use its internal structures (e.g. indexes and caches) to speed up query processing. Python code then generates a basic HTML view that can be further styled using CSS. It is a rather small codebase that – besides the Django framework – mainly relies on the SPARQLWrapper and RDFlib modules. The response times for entity pages with many links are several orders of magnitude faster than the prior "Pubby" installation.

As the code base is very compact and easy to understand, extended functionality can easily be added. Correspondingly, the HTML view has evolved quite a bit and now includes

- multiple CSS variants, including a "dark mode"
- provenance information on every statement to indicate the data source
- settings to limit the labels to one or more languages
- interactively expandable attribute sections that limit the number of values by default to keep the view compact

- filters to hide information contained in specific subsets of the data (separate graphs in the triple store)

The framework allows for the development of plug-in-like expansions that add new functionality to the server. As an example, we implemented a correlation analysis that uses the current page as a starting point. Another expansion is a configuration interface to define search indexes based on the RDF classes in the triple store. Admins can select the attributes and the indexing method, and this configuration data is then used to export the needed information as XML and feed it to an Elasticsearch instance. User-facing search interfaces can then be created automatically using the index configuration.

All software is available as open source and makes no assumptions on the data model that is the basis for the RDF data. It is also independent of the triple store software and uses nothing but the SPARQL interface. We consider it applicable for any research project that plans to make their data available as Linked Open Data and need a lightweight and fast web interface.

Current development is focussed on making the advanced search configuration process easier. We are also working on a more user-accessible start page.

Bibliography

- [1] Pfeffer, Magnus; Roth, Martin (2020): Japanese Visual Media Graph: Providing researchers with data from enthusiast communities. In: 2019 Proceedings of the International Conference on Dublin Core and Metadata Applications, ed. Koraljka Golub, Marcia Zeng, Paul Walk, Sam Oh and Tom Baker, 136–41. Seoul, Korea: Dublin Core Metadata Initiative (DCMI). Available at <https://dcpapers.dublincore.org/pubs/article/view/4259/2453>.
- [2] Pfeffer, Magnus; Kacsuk, Zoltan; Roth, Martin (2022): Japanese Visual Media Graph - Bündelung des Wissens von Fan-Gemeinschaften in einem domänenspezifischen Knowledge Graph. In: DHd 2022 Kulturen des digitalen Gedächtnisses. 8. Tagung des Verbands "Digital Humanities im deutschsprachigen Raum" (DHd 2022), Potsdam, ed. Michaela Geierhos, Peer Trilcke, Ingo Börner, Sabine Seifert, Anna Busch, Ulrike Wuttke, Melanie Seltmann and Kristina Genzel, 151–55. Zenodo. DOI:10.5281/zenodo.6304590.
- [3] Kiryakos, Senan; Pfeffer, Magnus (2021): The Benefits of RDF and External Ontologies for Heterogeneous Data: A Case Study Using the Japanese Visual Media Graph. In: Information between Data and Knowledge. Information Science and its Neighbors from Data Science to Digital Humanities. Proceedings of the 16th International Symposium of Information Science (ISI 2021) Regensburg, Germany, 8th–10th March 2021., ed. Thomas Schmidt and Cristian Wolff, 308–20. Glückstadt: Verlag Werner Hülsbusch. DOI:10.5283/epub.44950.
- [4] Baierer, Konstantin; Dröge, Evelyn; Eckert, Kai; Goldfarb, Doron; Iwanowa, Julia; Morbidoni, Christian; Ritze, Dominique (2017): DM2E: A linked data source of digitised manuscripts for the digital humanities. In: Semantic Web, 8 (5), 733–45. Available at <http://www.semantic-web-journal.net/system/files/swj1299.pdf>.
- [5] <https://github.com/LodLive/LodView>
- [6] <https://cliopatria.swi-prolog.org/home>