Knowledge Graphs

Project 13A. Transform any static dataset to a knowledge graph with RML

Preliminary Report

Illia Tesliuk, 296076

The Problem

This project aims to achieve the transformation of structured tabular data, where rows represent observations and columns represent attributes for these observations, into a knowledge graph represented by RDF subject-predicate-object triples. The goal of the transformation is to preserve relationships and semantics among different entities.

Motivations behind the problem

There are several motivations to convert static datasets into knowledge graphs. Firstly, contrary to tabular data, knowledge graphs allow for the representation of complex relationships between entities, thus enabling the discovery of connections that may not be apparent in tabular formats. Secondly, with structured connections between entities, knowledge graphs make it easier to perform sophisticated search queries. Finally, knowledge graphs facilitate the integration of data from various sources and formats, offering a unified framework for the representation and organization of information.

Tools

One of the common ways to transform static data into RDF graph is to use RML (RDF Mapping Language). RML is a language specifically designed for expressing mappings between different data formats, such as CSV, XML, and RDF) graphs. It provides a way to define how the data in these non-RDF formats should be mapped to RDF triples, which consist of subject-predicate-object statements.

However, RML mappings appear to have been designed with a focus on machine consumption rather than human comprehension. Therefore, YARRRML is a more common choice to create transformation mappings. YARRRML is a human-readable text-based representation for declarative Linked Data generation rules. A user defines mapping rules in a YAML file and YARRRML converts them into an RML mapping which can be later used to create RDF triples.

Dataset

For this project I've selected an open Spotify Songs database from Kaggle (https://www.kaggle.com/datasets/joebeachcapital/30000-spotify-songs). It consists of a CSV file with almost 30000 records with the most basic information about the tracks, namely, its name, artist, album name, duration, playlist, genre. Additionally, the dataset provides floating-point metrics, such as danceability, energy, tempo, etc.

However, this dataset requires some preprocessing before being mapped into RDF triples. Namely, some tracks involve multiple artists, which is usually indicated via 'feat' keyword in either track name or artist name. It would be difficult to create a custom parse function in YARRRML, therefore this preprocessing must be done in Python.

I've detected 3 types of entities to be extracted from the Spotify dataset – tracks, artists, and playlists. Therefore, the initial CSV file was split into 3 separate tables. This had to be done in order to enable a convenient linking process between there types of entities at the mapping stage.

Ontologies

I didn't manage to find a single ontology that would cover the majority of the track attributes presented in the dataset. I'm using Music Ontology Specification (http://musicontology.com/specification/) to describe such terms as 'Track', 'MusicArtist' and feature as 'duration'. Additionally, I've selected the Playlist Ontology (https://lov.linkeddata.es/dataset/lov/vocabs/plo) to describe the 'Playlist' class.

Since the floating-point metrics described above are unique to the Spotify dataset and are not present in any music-related ontologies, I've decided to create a custom Spotify ontology to cover these terms.

Source Code

Initial work and example outputs have been published to GitHub repository (https://github.com/IlliaTesliuk/2023Z-KnowledgeGraph-Project)