KGTK: A Toolkit for Large Knowledge Graph Manipulation and Analysis

Abstract of your accepted paper:

Knowledge graphs (KGs) have become the preferred technology for representing, sharing and adding knowledge to modern AI applications. While KGs have become a mainstream technology, the RDF/SPARQL-centric toolset for operating with them at scale is heterogeneous, difficult to integrate and only covers a subset of the operations that are commonly needed in data science applications. In this paper we present KGTK, a data science-centric toolkit to represent, create, transform, enhance and analyze KGs. KGTK represents graphs in tables and leverages popular libraries developed for data science applications, enabling a wide audience of developers to easily construct knowledge graph pipelines for their applications. We illustrate KGTK with real-world scenarios in which we have used KGTK to integrate and manipulate large KGs, such as Wikidata, DBpedia and ConceptNet, in our own work.

Link to a public github/bitbucket/gitlab repository:

https://github.com/usc-isi-i2/kgtk/

A copy of the README file from your repository, containing information on the platform required to run your code.

Readme file can be downloaded at https://github.com/usc-isi-i2/kgtk/blob/master/README.md (or see additional materials). Note that additional documentation may be found at: https://kgtk.readthedocs.io/en/latest/

Information on how to obtain the data needed to evaluate your submission. The data must be made openly available:

KGTK is a framework for manipulating knowledge graphs. In order to test the framework, you can use your own RDF data (after converting it to KGTK format with the import-ntriples command); or use any of the notebooks and examples provided in the /examples directory. The example directory (https://github.com/usc-isi-i2/kgtk/tree/master/examples) contains 8 Jupyter notebooks. The notebooks:

- commands/text_embeddings.ipynb
- CSKG Use Case.ipynb
- Example1 - Embeddings.ipynb
- Example2 - Curation and Statistics.ipynb
- Example3 - Reachability.ipynb
- Example5 - AIDA AIF.ipynb

Contain references to the data used (pointers are provided in the notebook itself, or the reference to the data is available in /examples/sample_data). Processing these notebooks will not take significant amounts of time (except perhaps the steps calculating embeddings)

The notebooks:

- Example4 - Wikidata Pagerank.ipynb
- Example6 - Wikipedia Tables.ipynb

Load datasets that are not in the repository. These datasets can be found, for Example4 in:

(16GB) https://drive.google.com/file/d/1WIQIYXJC1IdSlPchtqz0NDr2zEEOz-Hb/view?usp=sharing

(886 MB) https://drive.google.com/file/d/1m4x3Wpl8armvao6RCWlNyVT_IfpdHhHJ/view?usp=sharing

And for Example5, the datasets are available in:

https://github.com/bfetahu/wiki_tables_kg/#resource-overview

Processing notebooks 4 and 5 will take a significant amount of time and space. We recommend testing the framework based on the first batch of notebooks indicated above.

Other releases for KGTK Wikidata files are available in:

https://github.com/usc-isi-i2/wikidata-kgtk-releases

Environment

Most of these questions are already answered in the documentation of the tool and the readme file. Here we will add only the pointers to such documentation.

How to obtain the system? Follow the instructions listed on the readme file. You can use the binder examples to run specific notebooks; use the Docker images or follow the instructions to install the requirement files yourself in a new conda environment. All these steps are described in detail in the readme file.

How to configure the environment if need be (e.g., environment variables, paths)? N/A

How to compile the system? (existing compilation options should be mentioned and why they are needed)

KGTK uses Python 3.6 or higher. The instructions to run the system (no compilation is needed) are listed in the readme.md file. KGTK has been tested in Unix; and in Windows through Docker.

How to use the system? (What are the configuration options and parameters to the system and where to set them?)

In order to test that KGTK has been properly installed, you should type the following command:

```
kgtk -h
```

If the list of available commands is shown, then KGTK has successfully been installed. To use the different commands, please have a look at the documentation and examples: https://kgtk.readthedocs.io/en/latest/

Tests

Tests are not needed for making KGTK run. However, we have unit tests at:

kgtk/kgtk/tests/

Tests may be run with the following command (installing the requirements-dev.txt is necessary):

cd kgtk/tests

python -W ignore -m unittest discover

The open license under which you plan to provide the source code and data.

MIT License