Contents

1 Ontozeolite KG preparation

1.1 Quick start

To instantiate a copy of ontozeolite knowledge graph one needs:

- the input data (stored in directory `ontozeolite`),
- python code (in directory `python`),
- control scripts (`*.bat` files in the root directory).
- a running copy of Blazegraph database on a server with an empty namespace.

The data generation requires less than 10 Gb of hard drive space.

The code uses several external packages. It is recommended to use a virtual environment to install them:

```
$ python -m venv <venv_name>
$ <venv_name>\Scripts\activate.bat
(<venv_name>) $
```

Install third-party package `pymatgen`:

```
(<venv_name>) pip install pymatgen$
```

More information can be found at their official web-site.

Install third-party package `bibtexparser`.

The BibtexParser library requires version 2+. It has to be loaded from development branch, and NOT from 'pip install'. Pip install currently has version 1.3 or 1.4. Command line to install:

```
(<venv_name>) pip install --no-cache-dir --force-reinstall
        git+https://github.com/sciunto-org/python-bibtexparser@main
```

More information can be found at their official web-site.

Install Third-party package `entityrdfizer`:

```
(<venv_name>) $ pip install entityrdfizer
```

More details on the TWA web-site.

Install Third-party package `pyuploader`:

```
(<venv_name>) $ pip install pyuploader
```

More details on the TWA web-site.

Before instantiation change the SERVER and NAMESPACE variables in file `ontozeo.bat` to a valid server address and an empty namespace on that server. Add a password file for a server, if the server requires authentication: a file `blazedev.auth` in the parent directory must contain one line: `username:password`.

Now the entire KG generation can be done by a single command:

```
ontozeo.bat
```

The individual steps used in this script are described below.

Once fully uploaded, the KG can be queried by SPARQL queries or programmatically. Example SPARQL queries can be found in `ontozeolite/queries/`.

1.2 Overview

The zeolite knowledge graph (KG) comprises interconnected entities derived from various ontologies. The structure of the ontology can be found in the manuscript. These entities are instantiated from input data using different parts of the code, as described below.

The entire data for the zeolite KG is divided into parts according to the nature of the data:

A. Bibliography information. Uses BibTEX file(s) as input data. Output is `onto_bib` KG,

B. Crystal information. Uses Crystallographic Information Files (CIF) as input. Output is `cif_twa` KG,

C. Zeolite-specific information. Uses various input data in `.json` or `.csv` format, IRIs defined in `onto_bib`, `cif_twa` and some other external ontologies. Output is `ontozeolite_kg` KG.

Instantiation of the zeolite KG on a Blazegraph server consists of:

1. Preparation of input data,
2. Generation of CSV files,
3. Generation of OWL files,
4. Uploading the data to Blazegraph server,

The default directory for the data is `ontozeolite`. The file structure:

```
ontozeolite/biblio/bibfiles/  - input data (required)
ontozeolite/biblio/csv/    - generated, temporary file
ontozeolite/biblio/owl/    - generated, to upload
ontozeolite/crystal/data/  - input data (required)
ontozeolite/crystal/csv/  - generated, temporary files
ontozeolite/crystal/owl/  - generated, to upload
ontozeolite/zeolite/data/ - input data (required)
ontozeolite/zeolite/csv/  - generated, temporary files
ontozeolite/zeolite/owl/  - generated, to upload
```

1.3 Bibliography Information KG

Input:

`ontozeolite/biblio/bibfiles/` - individual bib file(s) (one citation per file),

`ontozeolite/biblio/bibdata_crossref_doi.tex` - a list of bibtex entries,

`ontozeolite/biblio/bibdata_original_pdf.tex` - a list of bibtex entries.

Processing:

```
python combine_bib.py

python bib2csv.py

csv2rdf ontozeolite/biblio/csv/onto\_bib.csv --csvType=abox
```

Output:

`ontozeolite/biblio/csv/onto_bib.csv` - file containing bibliography information in csv format,

`ontozeolite/biblio/owl/onto_bib.owl` - OWL file with all bibliography information, converted from `onto_bib.csv` (see above), To be uploaded to the Blazegraph server.

`ontozeolite/biblio/bib_iri_list.csv` - list of bibliography items and the corresponding IRI used in the `onto_bib.csv` file. This file will be used to link ontozeolite ontology to the bibliography information.

The OWL file for the bibliography part of the KG is generated from the standard BibTeX bibliography file(s). Each bibliography entry is stored as an entity of `bibo:Document` class. The TBox for `bibo:Document` can be found, for example, in documentation folder:

`ontozeolite/docs/20210503_ProvenanceOntologies_jb2197.pptx`

1.4 Crystal Information KG

Input:

`a_final_species_nodup.json` - a list of zeolitic materials, only CIF files mentioned in this list produce abox.

`cifdir/` - directories with CIF files for materials to be processed,

`CIF` - CIF files for zeolite frameworks.

Processing:

```
python crystalinfo.py

csv2rdf ontozeolite/crystal/csv/cif\_twa\_i.csv --csvType=abox
```

(where i=0...128).

Output:

`cif_twa_i.csv`, (where i=0...128),

`cif_twa_i.csv.owl`, (where i=0...128),

`cif_iri_list.csv`.

The total size of the ABox for crystal information for the zolitic materials is approximately 3.0 Gb. Due to limitations of the uploader the data is divided in separate files not exceeding 50 Mb.

1.5 Zeolite KG

There are currently 256 zeolite frameworks and over 1000 materials, each material belongs to a framerowk. The file size for the KG containing these frameworks and materials is close to 100Mb, so the data is separated in 3 parts with 100, 100 and 56 frameworks, respectively.

Input:

```
a_final_species_nodup.json
ontozeolite/zeolite/data/*.*
cif_iri_list.csv
bib_iri_list.csv
```

Processing:

```
python csv_maker.py -c all -f 0 -t 100 -o dir
python csv_maker.py -c all -f 100 -t 200 -o dir
python csv_maker.py -c all -f 200 -t 300 -o dir
python csv_merger.py
csv2rdf ontozeolite/zeolite/csv/ontozeolite\_kg\_i.csv --csvType=abox
```

Output:

`ontozeolite_kg_0i.csv` (here i=0,1,2).

`cif_iri_list.csv`

1.6 Generation of OWL files

OWL files are created from CSV files using rdtfizer tool: https://github.com/cambridge-cares/TheWorldAvatar/tree/main/EntityRDFizer After activating the virtual environment for each csv file run:

```
csv2rdf path/to/csv/file.csv --csvType=abox
```

1.7 Upload OWL files to `Blazegraph`

All upload in done by a single script:

`upload_cryst.bat`

The upload is implemented using `https://github.com/cambridge-cares/TheWorldAvatar/tree/`

It allows upload either to a local Blazegraph server, or remote with authentication.