FuseMine

Overview

FuseMine is a comprehensive pipeline designed to link data across multiple mineral site databases, such as USMIN and MRDS, and mineral site records extracted from reports and tables through tools created by TA2 Inferlink and USC. It facilitates the reconciliation process by utilizing both geospatial data and semantic similarities in textual attributes, such as site name and commodity.

FuseMine offers two primary functions: data processing and data reconciliation. In the data processing phase, raw structured data is converted into a standardized schema compatible with the Knowledge Graph (KG), where TA2 stores all processed data. During the data reconciliation phase, FuseMine queries the KG based on the commodity and links records representing the same mineral site.

Contents:

Installation

1. **Clone the FuseMine GitHub Repository**:

Unset

```
git clone
https://github.com/DARPA-CRITICALMAAS/umn-ta2-mineral-site-linkage.git
cd umn-ta2-mineral-site-linkage
```

2. **Memory Allocation**:
   The memory allocation requirement for FuseMine depends on the size of the data. However, FuseMine recommends allocating at least 10GB of memory for any task.

3. **Environment Setup:**
   FuseMine requires Python version 3.10 or higher and PyTorch version 2.0 or higher. To create a Conda environment named ‘**`fusemine`**’ with Python 3.10, run the following commands:

Unset

```
conda create -n fusemine python=3.10
conda activate fusemine
```

4. **Install PyTorch:**
   Installation instructions for PyTorch can be found here. While CUDA support is highly recommended it is not required to run FuseMine.

5. **Install Required Packages:**
   To install the necessary package libraries, execute the following command:

Unset

```
pip install -r requirements.txt
```

Note: Both the data processing and data reconciliation steps in FuseMine require an active network connection.

Usage

Data Processing

FuseMine reconciliation relies on data available in the MinMod Knowledge Graph (MinMod KG). To include new data in the linkage process, raw databases must first be processed. FuseMine includes a raw data processing tool compatible with structured databases such as MRDS and USMIN.

The data processing step requires a manually curated attribute map file, which maps the headers of the raw structured data to the MinMod KG schema. An example of an attribute map CSV is provided: **`sample_mapfile.csv`**.

After creating the attribute map CSV, use the following command to process the data:

Unset

```
python3 process_data_to_schema.py --raw_data <path_to_raw_CSV> --attribute_map
<path_to_attribute_map> --schema_output_directory <path_to_output_directory>
--schema_output_filename <output_file_name>
```

Description of Arguments:

`--raw_data`: Path (either file or directory) to the raw mineral site database.
`--attribute_map`: Path to the CSV file containing label mapping information (refer to `sample_mapfile.csv` for guidance)
`--schema_output_directory`: Directory where the processed database will be saved.
`--schema_output_filename`: Filename for the processed database.

Data Reconciliation

Before running FuseMine, ensure that all necessary data is available in the MinMod KG. You can check the last update time and version of the MinMod KG here.

To perform data reconciliation using distance-based intralinking and area-based interlinking methods, execute the following command:

Unset

```
python3 fusemine.py --commodity <commodity_name> --intralink distance
--interlink area
```

Description of Arguments:

`--commodity`: The specific commodity to focus on. The commodity name must match one of the MRDS commodity names listed here.
`--single_stage`: Method to use for location-based single-stage linking (options: ‘**`distance`**’ / ‘**`area`**’)
`--intralink`: Method to use for location-based intralinking (options: ‘**`distance`**’ / ‘**`area`**’)
`--interlink`: Method to use for location-based interlinking (options: ‘**`distance`**’ / ’**`area`**’)
`--same_as_directory`: Directory where the output “same as” CSV files will be stored (default: **`./output`**)
`--same_as_filename`: Filename for the “same as” CSV file (default: **`./<commodity>_sameas.csv`**)

FuseMine generates a log file that reports the number of available records on the MinMod KG, the number of identified mineral sites, and the run time. These log files are saved in the **`./logs`** directory.

Evaluation

To evaluate FuseMine’s performance on Tungsten assessment data from the Idaho/Montana region (Goldman et al., 2020), run the following command:

Unset

```
python3 fusemine.py --tungsten
```

The evaluation provides group-wise accuracy and link-wise F1 scores, compared against a baseline method that uses only location data.

Data Upload

To ensure that the outputs of FuseMine, whether processed raw data or linkage information, are reflected in the KG, they must be uploaded to the TA2 data repository and merged into the main branch.

File Organization:

- Place all processed raw data in the `umn` folder.
- Place all linkage information in the `sameas` folder.

Adding Additional Data:

- To add a new mineral site record or linkage information upload the new file to the appropriate folder in the repository.
- For detailed steps on adding a file to the repository, refer to the instructions on the official GitHub documentation.

Updating Existing Data:

- To update existing data on the KG, delete the previous file from the repository and then upload the updated file.

Once the data is merged into the main branch, it typically takes 5 to 10 minutes for the changes to be reflected on the KG.