Information Retrieval and Extraction

Minimum Viable Product Report for Major Project # 2

Project

Create a high-Quality Database for a specific domain using Wikipedia and external sources.
Domain: Minerals

Team Members

- Pratyush Pratap Priyadarshi (2019101118)
- Triansh Sharma (2019101006)
- Rahul Khandelwal (2019900037)
- Arvin Goyal (2019900039)

Why Domain Mineral

A Mineral is a natural substance with distinctive chemical and physical properties, composition, and atomic structure. Minerals are basic and essential raw materials in our daily lives and are vital for economic, social, and technological development. They are non–renewable natural resources that are vital for the construction, manufacturing, and energy industries and need to be utilized in an optimum way.
In this project, we are trying to create relational data about various attributes of minerals that can be utilized across different applications or industries.

Aim

As part of this project, we aim to create a high-quality database of minerals. This database will be based on information available in the Wikipedia ecosystem and various structured and unstructured external sources.

Methodologies

Getting data from the Wikipedia ecosystem

SEED URLs

1. List of minerals
2. Classification of silicate minerals
3. Classification of non-silicate minerals

1. As the first step, we intend to collect all minerals data present in the Wikipedia ecosystem. Primarily we chose to extract attributes from the **infobox** of Wikipedia mineral pages.
2. We explore Wikipedia programmatically to figure out what all pages in Wikipedia are about minerals using the **wikipedia** library. We also made use of categories in Wikipedia for this purpose and focus specifically on the **Category: minerals**. We prepare a list of minerals and their Wikipedia pages to be used downstream.

3. We parse these Wikipedia pages programmatically to collect infobox present on each page. The information available in infoboxes is stored as key-value pairs, retrieved using the **wptools** library and stored in a CSV file.

Getting data from external sources

Webminerals:

1. Webmineral is a mineralogy database that has a collection of nearly 4.6k minerals.
2. We collected the links of mineral pages present on the web mineral site.
3. We wrote a script using **selenium** which goes over each page link, performs HTML parsing, and then scrapes key-value pairs from the text.
4. In the end, we obtain a JSON file containing information about minerals, which is further converted into a CSV file.

International Mineralogical Association (IMA) List of minerals

1. IMA provides a list of minerals in a form of a PDF. The pdf is present in the data directory of the repository.
2. We use **tabula-py** to extract tables from the PDF and dump the data into a CSV file.

Data cleaning

1. Many times during scraping, we found out there are a lot of redundant attributes that came along due to the presence of data in an inappropriate format. Hence Attributes fetched so far are sparse.
2. Examples include the absence of infoboxes in Wikipedia pages, unformatted text in tables from external sources, etc.
3. In order to create a denser database, we consider attributes having a **density of 40% or more** across all minerals gained from that particular source. We also tried to merge similar attributes to create a more compact and dense database, but we need to work upon this part further.

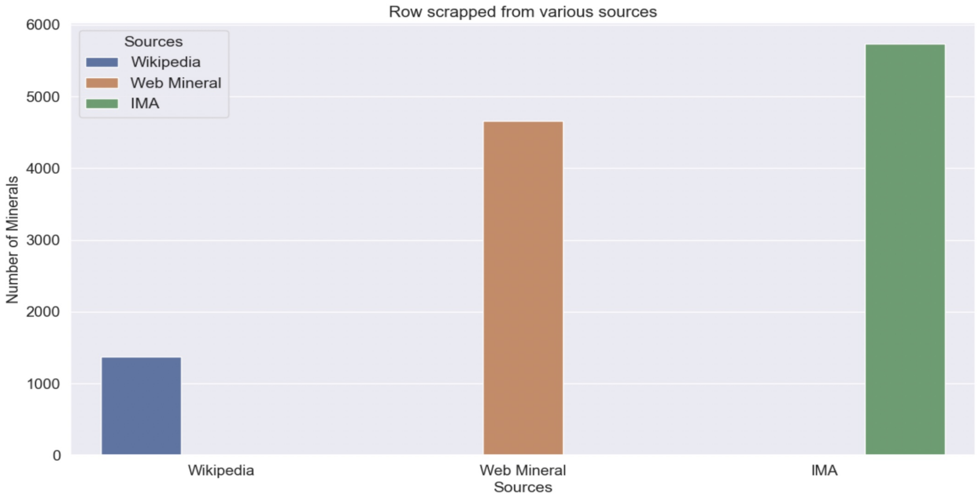
4. All data conversions and cleaning processes are done using **regular expression** and the famous **pandas** library in python.

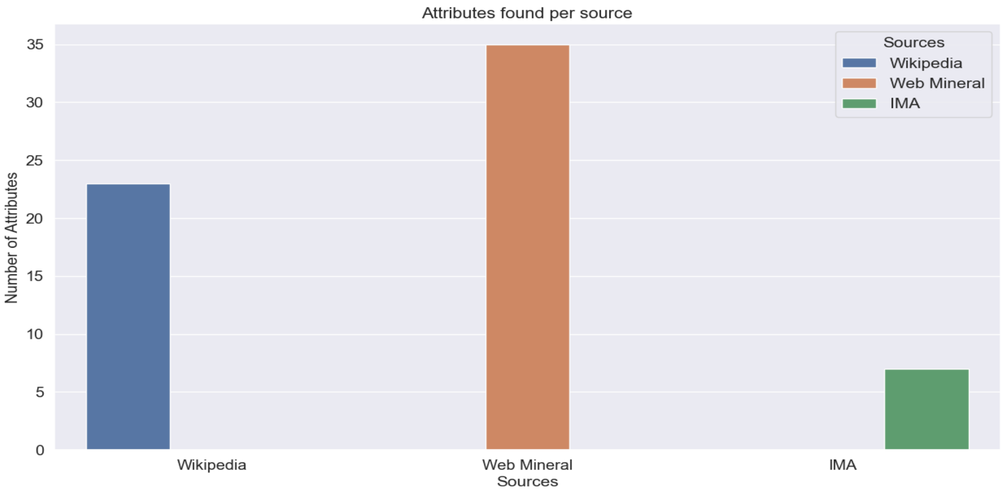
Findings

After scraping data from these sources, we collected the following information.

Source	Number of rows	Attributes
Wikipedia	1371	23
Webmineral	4660	35
IMA list of mineral	5739	7

Note: A few of the attributes might be common among these three sources.





Observations

1. We came to know that mineral is not a very common domain, there is not a lot of data present on Wikipedia pages. Although the categories in Wikipedia provide a long list of minerals, on further examination, we found out that the articles present in Wikipedia are not properly formatted and there is very little and sparse content. For many minerals, Wikipedia pages do not exist (empty pages with red links) and a lot of them don't contain infoboxes (information is present in the form of plain text).
2. The mineralogy databases like Webmineral and Mindat are cohesive and densely populated with information in the form of tables.

Repository

- **Link to the repository:** https://github.com/Triansh/Domain-specific-data-collection
- The CSV files from each source are present here.
- The code is divided into 3 directories:
    - Crawlers - Contains scripts to find out links and mineral names from Wikipedia
    - Scrapers - Contains scripts to scrape data from Webmineral, IMA pdf and Wikipedia.
    - Cleaners - Scripts used to apply a threshold to data frames, JSON to CSV conversions, and minor cleaning methods.
- All the data obtained is stored in the data directory
- Scripts to create plots and analyze data are present in the Plots directory.

Where are we w.r.t Scope Document

1. We are aligned with our scope document and there is hardly any deviation in terms of timeline and work package.
2. We choose a domain of minerals that has around 5000 types with around 50 attributes.
3. We have already collected data from Wikipedia and data from a third party (Webmineral) has been collected.
4. An effort towards making the database compact and dense has already started.

Future plan

1. To gather more data from the Wikipedia ecosystem, we plan to use DBpedia/ Wikidata to collect more meaningful attributes not gained so far. They both act as structured sources and expect us to provide queries in the form of RDF triples.
2. Mindat acts as another widely used mineralogy database run by the not-for-profit Hudson Institute of Mineralogy. We noticed it has a few more attributes that have not been retrieved from currently scraped sources like the reaction of minerals under UV rays.

3. Currently, the data retrieved from various sources are distributed in multiple files and not cleaned. The final goal is to prepare a single JSON/CSV file combining all the data from sources with proper merging (combining synonyms, British & American spellings etc) and removal of redundant attributes.