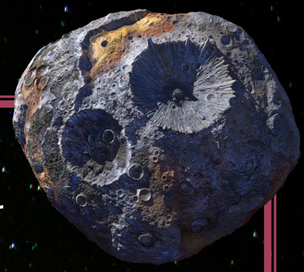
Iron Meteorite Database:
Automatic Human-Guided Data Extraction from Scientific Research Papers

Team: Kenneth Bonilla, Hajar Boughoula, Michael Falgien, Joshua Johnson, Troy Mullins.
Sponsor: Dr. Cassie Bowman, **Technical Mentor:** Dr. Devin Schrader

Ira A. Fulton Schools of Engineering
Arizona State University

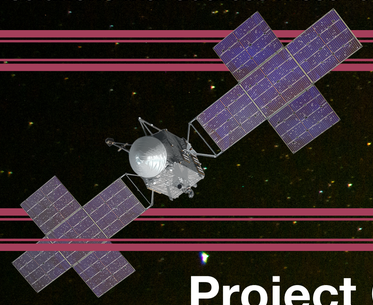


Background and Motivation

The Iron Meteorite Database is a capstone project that's part of the Student Collaborations component of NASA's Psyche Mission, led by ASU.

- The Psyche Mission is a journey to visit 16 Psyche, a metal-rich asteroid orbiting the Sun between Mars and Jupiter. The Psyche Spacecraft is targeted to launch in 2022 and arrive in 2026.
- To prepare for arrival at the asteroid, scientists would be aided by an easily accessible database of the major, minor, and trace element composition of iron meteorites.
- The data currently exist but are buried in scientific publications of various formats.

The Iron Meteorite Database implements a set of innovative solutions that address these requirements… and beyond!



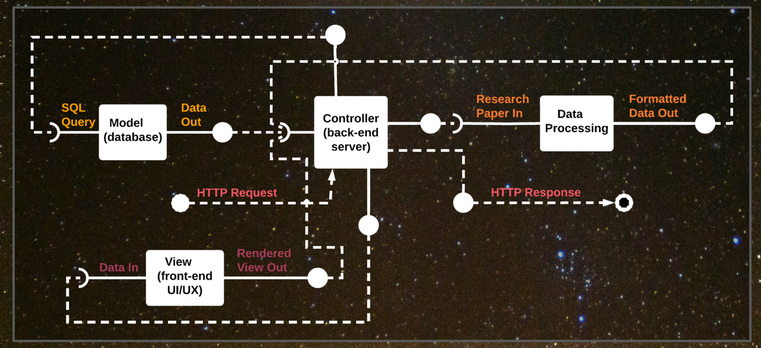
irondb.org

Project Overview

- Web Application features automatic human-guided processes to find, recognize, and collect the appropriate data from many different sources and deposit it into a comprehensive database.
- Python scripts assist scientists in data entry by automatically extracting information from research papers.
- User-friendly tool that allows scientists to easily search and export condensed data into standard scientific plots.

Technical Aspects

The Iron Meteorite Database implements a Model-View-Controller architecture (diagram below) leveraging an external module of scripts to provide tools for extracting element compositional data of iron meteorites from research papers and storing it into a format that is easy to search and export through a web app.



Creating the database:

- Using the application, data entry personnel can enter new data and manage existing data.
- Bootstrap and JQuery front-end provide interactive user interface.
- Web server built on Express and Node.js to handle requests.
- Postgres Database stores all collected data.

Extracting the data

- Tabula, pdfminer.six and PyPDF2 were used to stage PDF imports for data extraction and manipulation.
- External scripts implement innovative Natural Language Processing algorithms to extract paper attributes such as title, authors, source, and publishing year.
- Custom built algorithms extract table data with their journal page number, sort through false positives, and remove unwanted rows and columns.

Challenges

- How can code identify the title of a paper the same way a human does?

Hypothesis: The probability that a cluster of words constitutes the title of a paper is positively correlated with the weight of keywords extracted from the text body and which appear in said cluster of words above the body.

$$P(c_i = title) \approx \sum_{j=0}^{j=n(c_i)-1} weight(w_{i,j})$$

c_i = cluster of words
$n(c_i)$ = number of words in c_i
w = word in a cluster

- Finding which pages have tables on them .
- Determining if the tables that are found are relevant.
- Cleaning up table fields that contain erroneously extracted data.

Future Work

- Plotting extracted data with a tool that dynamically generates interactive graphs.
- Automatically detecting analysis techniques of each element in the paper text.
- Expanding data extraction algorithms to accurately classify and interpret research papers from any scientific field.
- Allowing the user to extract unpredicted attributes.
- Generate a summary that's a collection of summaries of each section of a paper.
- Refactoring server using a python-based web framework.

Artist's conception of asteroid and spacecraft: ASU/Peter Rubin, Background: NASA/Bill Ingalls