File Similarity by Code Representation : : **CHEAT SHEET**

About

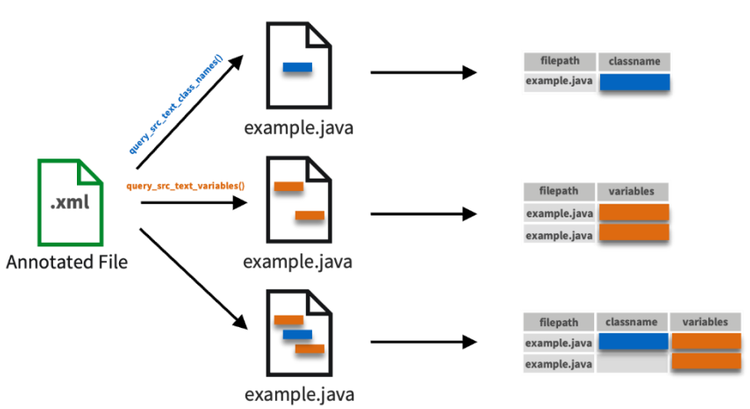
Files can be represented differently depending on how we subset their content, such as functions, class names, variables, or documentation. These representations shape the way files are similar to one another, as selected code snippets emphasize different semantic relationships between files. This cheat sheet describes how to use Kaiāulu to represent source code differently for semantic similarity.

Source Code Representation

The Syntax Extractor in Kaiāulu is used to extract selected content from source code using srcML. The srcML tool is used to annotate source code with descriptive tags that help in identifying code elements.

annotate_src_text(): generates an annotated XML file.

XPath queries are used to query the annotated file, and output a table of filenames and the source code element being queried (variable names, function names, file class documentation, etc.)



Query functions: Kaiāulu provides an API for extracting different portions of source code files as structured tables. These representations include variable names, function names, documentation, and more, each capturing specific file elements. Using these tables, we can analyze and transform code into selected representations for further processing.

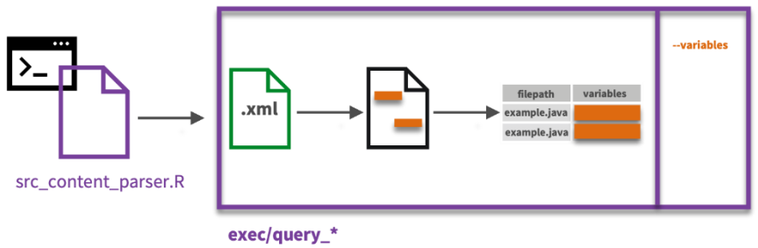
query_src_text_class_names(): outputs a table of with a class names column, and a file paths column.

query_src_text_namespace(): outputs a table with a namespaces, and a file paths column.

Relevant Notebook: syntax_extractor.Rmd

Exec Scripts

Exec scripts act as interfaces between Kaiāulu functions and external tools, exposing Kaiāulu's capabilities to external sources through the command line. Users can execute scripts for specific queries (e.g., extracting function names or documentation) from the terminal.

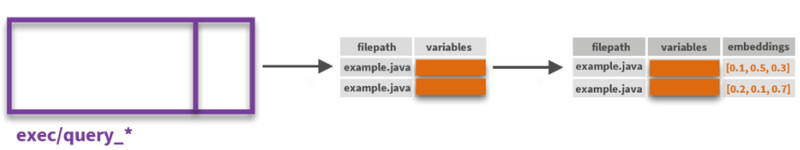


src_content_parser.R: serves as the main entry point for querying. It takes a flag specifying the query type (e.g., --class-names, --variables) and calls the associated query function. The annotated file is passed as a parameter, determining the contents to be queried.

query_* scripts: implement the logic for specific types of queries, such as extracting class names or variables. Each query focuses on a particular aspect of the code representation.

Embeddings

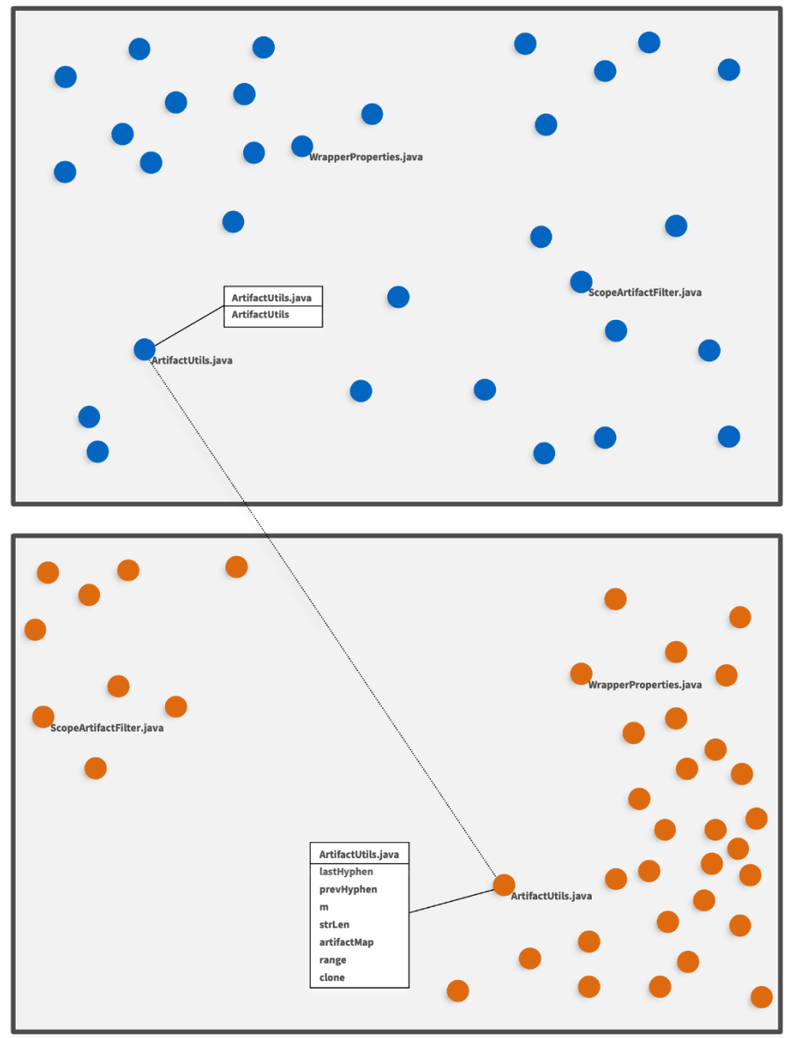
Word embeddings represent text semantics by mapping words or sentences to a numerical representation. The numerical vector representation can be used for semantic similarity comparisons which would be lost if direct word comparison was utilized.



Beyond the selection of a file representation, different models to convert content to embeddings can also be selected. For example, Java-specific FastText embeddings may be better for source code representations, while a file representation containing only code documentation may benefit from natural language embeddings instead of source code. Other language models may also be more appropriate if the source code is not Java. Kaiāulu parameterizes the pipeline.

UMAP

UMAP transforms high-dimensional embeddings by reducing dimensionality and fitting on a plane, which allows for visualization of the relationships between files.



The diagrams above show how the representation of code elements, such as class name and variables, impacts the relative positions of files in a UMAP projection. Each representation emphasizes different aspects of the code, resulting in varied groupings and relationships.

This variability demonstrates the importance of selecting the appropriate representation for the specific insights required. By choosing different code elements, similarity analysis can be tailored to answer diverse research questions.