Automating Semi-structured File Ingestion

A Global IDs Technical Whitepaper
January 14, 2020
info@globalids.com

This document is part of our series of educational technical whitepapers that we release to share our experiences and lessons learned in enterprise data management with a wider public.

Executive Summary

- *Ingesting information from partner databases* is a problem routinely faced by enterprises with many enterprise partners. Partner data is typically provided in semi-structured files in many different formats, with differing structures of attributes and field names.

- The data ingestion problem is difficult as well as universal. Solving it requires *intelligent mapping of partner input fields to target database columns*, a process that is often performed manually, even for large numbers of partner data sources.

- *Automated intelligent file ingestion* is possible, however, using the following core technologies:
    - *Metadata discovery* applied to both the target and the input sources.
    - *Machine learning classification* of source and target metadata.
    - *Automated mapping* of source to target attributes, followed by human verification.

- These core technologies overlap with those involved in *data lineage*. Implementing data lineage (described in a separate whitepaper) enables the solution of the data ingestion problem as a byproduct.

Background and use case examples

The ingestion of data from many partners or other external sources — the initial intake as well as ongoing updates — presents a difficult problem because of differences in input file formats and the lack of standardization in the data itself, down to the level of individual data attributes and attribute names.

Use case examples include:

- A group insurer's ingestion of employee subscriber data from potentially thousands of organizations holding group insurance policies.
- A retailer's ingestion of data from potentially thousands of product suppliers.
- A market data provider's ingestion of data from financial exchanges around the world.
- A credit monitoring organization ingesting data from many member organizations.

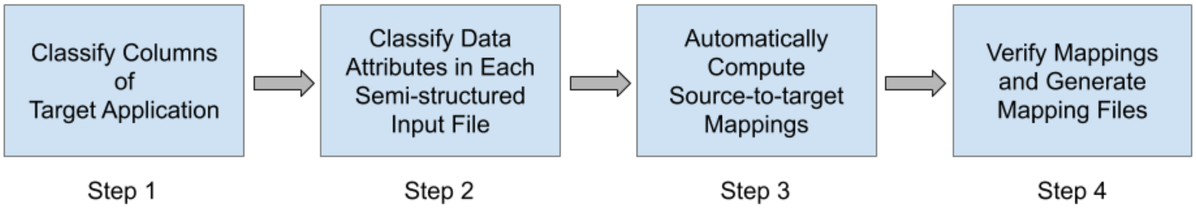
Data from external sources is typically provided in *semi-structured files* — often the result of internal database dumps — in a wide range of formats: comma-separated (CSV), Excel, XML, JSON, and many others. The only common denominator is machine readability: the existence of some procedure for parsing out items of a certain type (e.g., Employees) and their attributes (Employee ID, Name, Date of Birth, Employment Start Date, etc.).

Besides differing input file formats, the *representation* of data items — i.e., the number, order, and names of attributes — will also differ to some degree across all sources, and between each source and the target database. In order for data to be ingested into the target database, key data attributes in each input file must be *identified and mapped* to logically corresponding attributes in the target. This identification and mapping process is often performed manually, even for large numbers of input sources.

An automated solution for ingesting semi-structured files is thus highly desirable, since it eliminates substantial amounts of overhead as well as the potential for error associated with a manual ingestion process.

Overview of the automated process

This whitepaper outlines a *four-step process* for implementing automated ingestion of data from semi-structured files into a target application. The following figure depicts the four steps:


Classify Columns of Target Application
Step 1
Classify Data Attributes in Each Semi-structured Input File
Step 2
Automatically Compute Source-to-target Mappings
Step 3
Verify Mappings and Generate Mapping Files
Step 4

Step One: Analyze and classify attributes (columns) of target database tables. (*NOTE: This step needs to be performed only once.*)

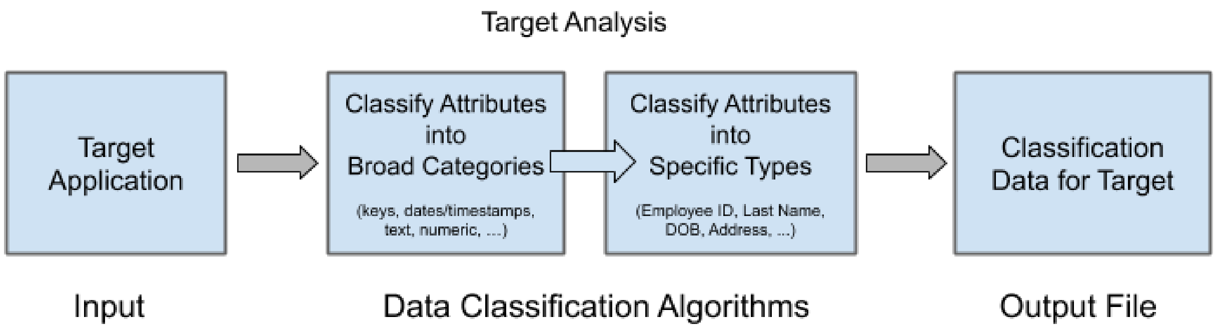
Step Two: Analyze and classify data attributes in each semi-structured input file.

Step Three: For each semi-structured input file, automatically compute the mapping from input data attributes to columns in the target database.

Step Four: For each semi-structured input file, verify the mapping and output a machine-readable mapping file usable by the target application for ingesting the data into its tables.

We now describe the four steps in more detail.

Step One: Analyze and Classify Data Attributes in Target Application


Target Analysis
Target Application
Classify Attributes into Broad Categories
(keys, dates/timestamps, text, numeric, ...)
Classify Attributes into Specific Types
(Employee ID, Last Name, DOB, Address, ...)
Classification Data for Target
Input
Data Classification Algorithms
Output File

In Step 1 we scan the target database for attribute (column) metadata pertaining to each relevant business concept. For example, a group insurer will need to ingest data from many sources that pertains to the concept of Employee Subscriber.

We then use machine learning classification software to perform an analysis of target attribute metadata in two stages:

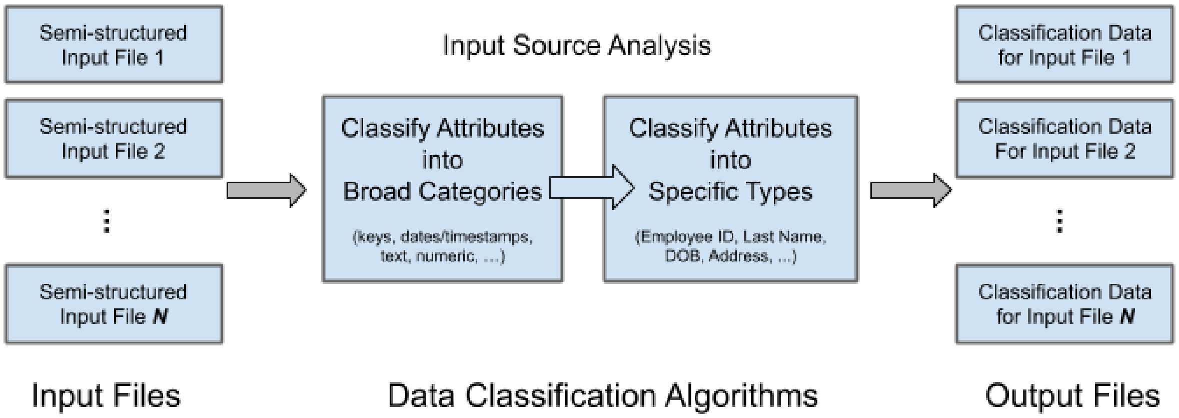
1A. The algorithms classify target attributes into *broad categories*. For example, keys, date and time fields, numeric fields, descriptive text fields, and so forth.

1B. The algorithms then further classify target attributes into *specific attribute types* relating to semantic domains linked to the business concept. For example, a key may be further classified as an Employee ID, a date field as Employee Date of Birth, a second date field as Employee Start Date, a descriptive text field as Last Name, a second descriptive text field as Street Address, and so on.

The output of Step 1 is an enriched metadata file containing the field classifications for target database columns related to each relevant business concept.

NOTE: As long as the target database schema remains constant, this first step in the process needs to be performed only once.

Step Two: Analyze and Classify Data Attributes in Each Semi-structured Input File


Semi-structured Input File 1
Semi-structured Input File 2
⋮
Semi-structured Input File N
Input Files
Input Source Analysis
Classify Attributes into Broad Categories
(keys, dates/timestamps, text, numeric, ...)
Classify Attributes into Specific Types
(Employee ID, Last Name, DOB, Address, ...)
Data Classification Algorithms
Classification Data for Input File 1
Classification Data For Input File 2
⋮
Classification Data for Input File N
Output Files

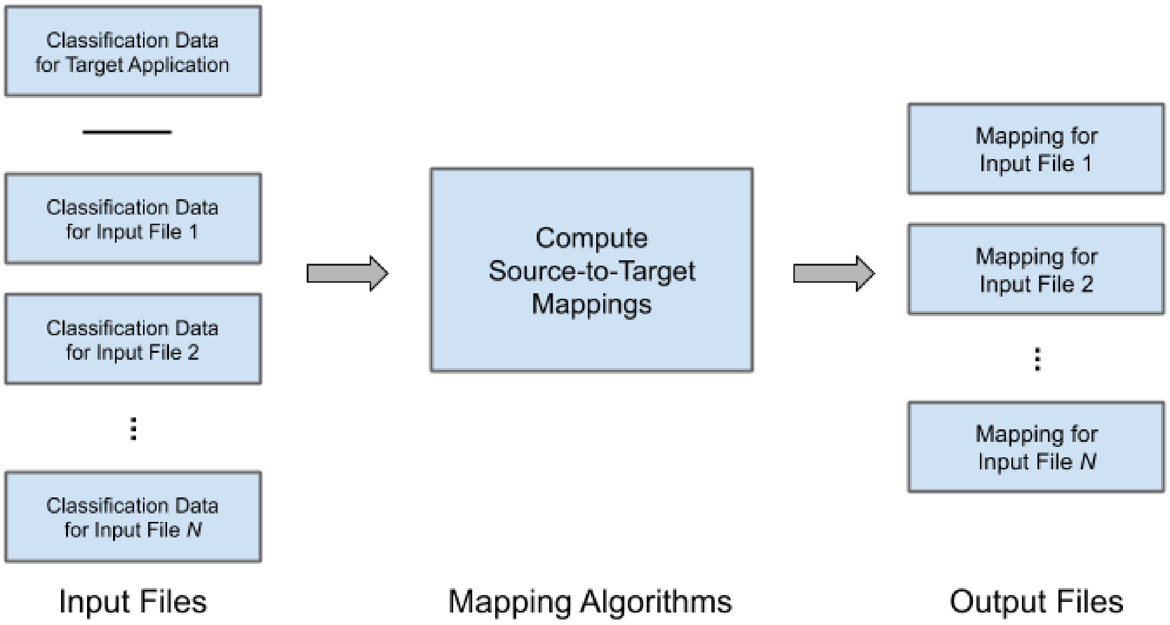
In Step 2, we scan each semi-structured input file for attribute metadata pertaining to the data items contained in that file.

We then use machine learning classification software to perform an analysis of source attribute metadata using the same two-stage process as for target attribute metadata:

> 2A. The algorithms classify source attributes into *broad categories*.
>
> 2B. The algorithms then further classify source attributes into *specific attribute types* relating to semantic domains linked to the business concept.

The output of Step 2 consists of an enriched metadata file for each input source that contains the classifications of the attributes found in the corresponding semi-structured input file.

Step Three: Automatically Compute Source-to-target Mappings for Each Input File

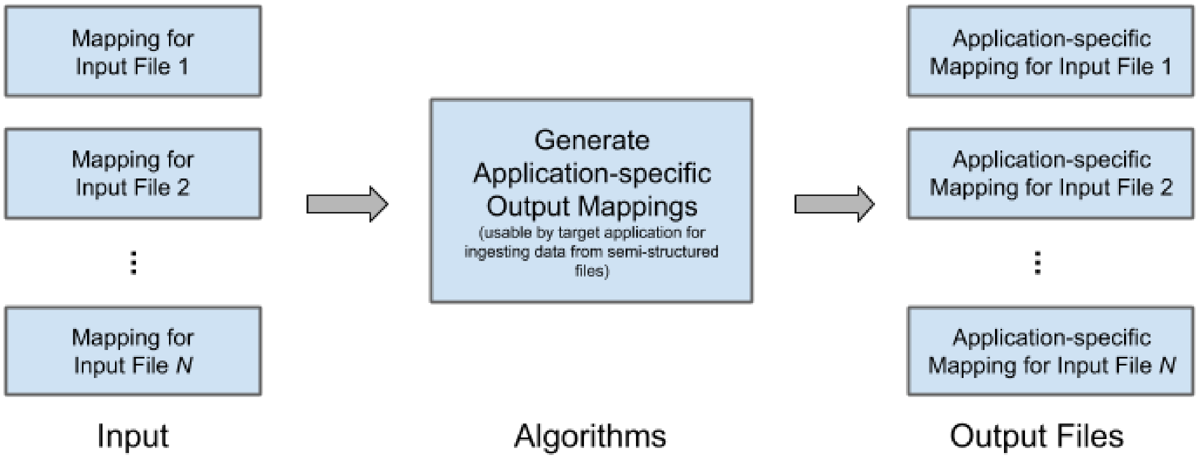

Classification Data for Target Application
Classification Data for Input File 1
Classification Data for Input File 2
⋮
Classification Data for Input File *N*
Compute Source-to-Target Mappings
Mapping for Input File 1
Mapping for Input File 2
⋮
Mapping for Input File *N*
Input Files
Mapping Algorithms
Output Files

In Step 3, given the classification data for the target and each input source, we can now automatically compute the mapping of input source attributes to target database columns for each semi-structured input file.

Each source attribute has been classified and linked logically to a key field of the relevant business concept (e.g., Employee Subscriber). The source attributes can therefore be *mapped automatically* to the corresponding attributes (columns) in the target database, regardless of any differences in the representation of attributes between source and target.

The output of Step 3 is a set of mappings linking the source attributes to the corresponding target attributes for each semi-structured input source file.

Step Four: Verify Mappings and Output Mapping Files for Data Ingestion


Mapping for Input File 1
Mapping for Input File 2
⋮
Mapping for Input File *N*
Generate Application-specific Output Mappings
(usable by target application for ingesting data from semi-structured files)
Application-specific Mapping for Input File 1
Application-specific Mapping for Input File 2
⋮
Application-specific Mapping for Input File *N*
Input
Algorithms
Output Files

In Step 4, a human verifies the source-to-target mappings generated in Step 3. This final step then outputs a machine-readable mapping file for each input source that ETL or similar functionality of the target application can use to ingest data from the input files.

When new data is received from the same external source, Steps 2 through 4 are repeated for each semi-structured input file. There is no need to repeat Step 1, unless the target database schema changes.

Conclusion

Ingestion of partner data from semi-structured files can be automated by machine learning classification of fields in the target application and each input source file, followed by mapping of input source fields to logically corresponding fields in the target application. After verification, mapping files may be generated for use by the target application for ingesting data from all input sources.

The core business concepts of particular enterprises are the logical drivers of the data ingestion process. For example:

- Group insurers will ingest data from policyholders pertaining to employee subscribers.
- Retailers will ingest data from suppliers pertaining to products.
- Market data providers will ingest data from exchanges pertaining to market transactions.

- Credit monitoring organizations will ingest data from data providers pertaining to consumers and businesses.

Each relevant core business concept (Employee subscriber, Product, Transaction, etc.) has core attributes for which a source-to-target mapping must be generated for every input source. This cannot be done without metadata classification of the target and each source, using advanced algorithms. Machine learning models can be trained for each business concept and attribute type to automate classification (Steps 1 and 2 above) and feed the results into a mapping algorithm (Step 3).

Once the classification and mapping infrastructure is set up for a given target application and a set of supported input file formats, all the overhead and error-prone nature of a manual process can be eliminated thenceforward. Ingesting data from semi-structured files into relational databases is just one example of a labor-intensive business process that can be fully automated using machine learning intelligence and suitable supporting algorithms.