PharmaSUG China 2025 - Paper AI-157

Leveraging AI for Efficient SDTM Variable Mapping: Semantic Matching Approaches

Huan Song, Guangan Cao, Jie Liu and Chunpeng Zhao, BeiGene

ABSTRACT

The Standard Data Tabulation Model (SDTM) mapping process is a critical yet time-consuming step in clinical data management, often requiring significant manual effort for identifying and mapping Case Report Form (CRF) variables to SDTM domains and variables. This paper explores the application of Artificial Intelligence (AI), specifically leveraging semantic matching techniques, to enhance and accelerate this process.

We present two distinct approaches implemented within the pharmaceutical data analysis environment.

The first approach utilizes Python, employing embedding models to generate semantic representations. CRF variables are described textually, and these descriptions are semantically matched against a library of previously mapped variables. High-similarity matches trigger the direct reuse of existing mappings, while lower-similarity cases prompt the AI to derive potential mappings based on related variables.

The second approach focuses on accessibility within the SAS ecosystem, demonstrating how semantic matching functionality can be integrated directly into SAS workflows by calling external AI service APIs. This eliminates the need for Python scripting within SAS environments. Both methods aim to reduce manual effort, improve mapping consistency, and expedite the SDTM mapping process, offering practical solutions for data managers seeking to leverage AI in their daily operations.

INTRODUCTION

Data collected via Case Report Form (CRF) often exhibits variations in explicit expressions while retaining identical semantic meanings. Traditional programming approaches typically struggle with such scenarios, frequently requiring manual inspection and judgment. Integrating AI's semantic capabilities into this decision-making process enables efficient adaptation of existing program logic. This significantly enhances textual adaptability, effectively overcoming the limitations imposed inherent in conventional programming techniques, particularly those relying on regular expressions. Consequently, tool applicability broadens, and mapping efficiency markedly improves.

Two primary methodologies are proposed to enhance the SDTM mapping.

The first approach integrates the embedding and chat modules to fully leverage existing Python-based knowledge repositories. For previously established mappings, the system directly adopts known solutions. For new fields identified through (CRF), the embedding module locates semantically similar knowledge and internal heuristics. This information refines and expand the prompt, enabling the chat module to generate contextually appropriate and plausible responses.

The second integrates database APIs from the DIFY platform with the PROC HTTP procedure in SAS. This integration facilitates seamless embedding within existing SDTM mapping codebases. Employing semantic matching optimizes prior SAS regular expression matching routines, supporting more comprehensive utilization of existing codes. Furthermore, it facilitates reference mapping retrieval and contributes to identifying suitable SDTM Controlled Terminology (CT).

USING EMBEDING AND CHAT MODEL TO GET MAPING RULE IN PYTHON

Commencing with a three-tiered hierarchical framework requires systematic categorization and annotation of fields collected via CRF: the CRF form, the field's label, and its parent label. A concise summary encapsulating these attributes serves as the foundational text for semantic matching.

Once fields are uniformly formatted per this schema, control logic augments direct semantic matching to mitigate AI-induced errors. When new fields fail predefined semantic criteria, knowledge-based prompts dynamically generated from contextual information instruct the GPT model to infer contextually appropriate outcomes.

WORKING FLOW AND EXPERIMENTAL DESIGN

The knowledge base is structured into two core components: a similarity computation module identifying semantically similar fields, and a foundational repository storing comprehensive information to enrich prompts. This supplementary context enables the AI model to generate accurate responses

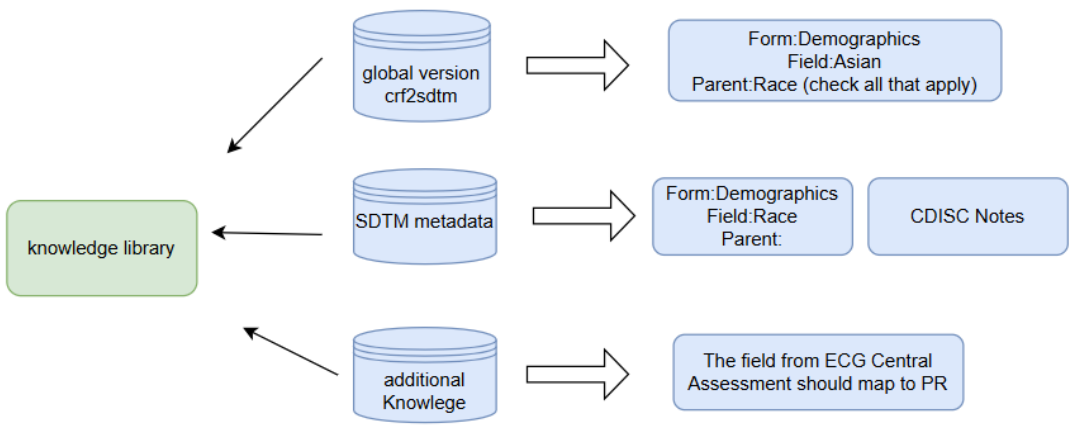

Figure 1 Structure of Knowledge Library

Embedding models facilitate retrieval of established methodologies from the knowledge base, ensuring high mapping consistency. Conversely, the chat module effectively handles novel fields. This capability is enhanced when the embedding model identifies prior strategies for similar fields, enabling their use to optimize prompt design for the chat module. Such optimization improves response accuracy and reasonableness. Furthermore, general principles or patterns not easily identifiable at the field level can be explicitly integrated into prompts, ensuring critical contextual information informs the AI's final inference.

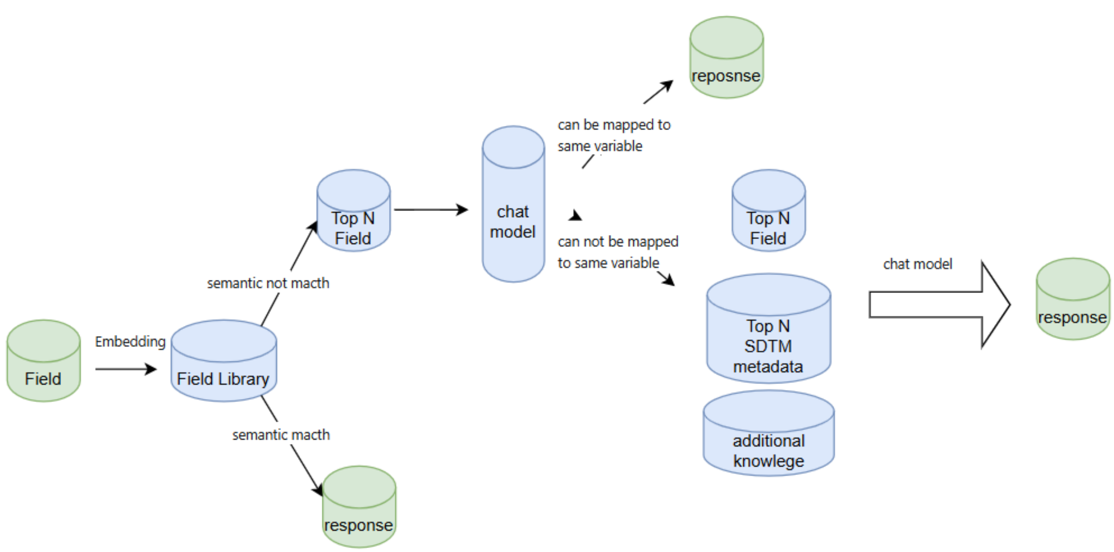

Figure 2 Workflow of mapping in Python

KEY FEATURES

- EFFECTIVE UTILIZATION OF HISTORICAL KNOWLEDGE.
- POWERFULL KNOWLEGE REPOSITORY OF AI
- FREELY UPDATED AND MAINTAINED INHOUSE KNOWLEGE BASE

USING EMBEDING MODULE IN SAS MACRO

Within conventional SAS-based SDTM mapping workflows, a key challenge involves distinguishing between semantically equivalent field expressions. Furthermore, maximizing the reuse of accumulated historical programming expertise is critical. Integrating embedding models directly addresses both limitations. By encapsulating embedding generation and similarity calculation within a SAS macro, this method preserves established workflows while overcoming inherent constraints of traditional regular expression matching, significantly enhancing mapping accuracy.

WORKING FLOW AND EXPERIMENTAL DESIGN

As an AI-driven conversational agent platform, DIFY provides database management APIs. Integrating these DIFY APIs with SAS PROC HTTP enables efficient construction of vector databases and execution of database queries. This integration streamlines development processes, allowing rapid prototyping and deployment of advanced data handling capabilities within SAS.

```
%let json_payload = {
  "name": "&file_m.",
  "text": "&text",
  "indexing_technique": "high_quality","process_rule": {"mode":
"automatic"}
};

filename jsonfile temp;
data _null_;
  file jsonfile;
  json_payload = symget('json_payload');
  put json_payload;
run;

filename resp temp;
proc http
  url="https://api-
dify.beigenecorp.net/v1/datasets/&libid_m/document/create-by-text"
  method="POST"
  in= jsonfile
  out=resp;
  headers
    "Authorization"="Bearer &token"
   "Content-Type"="application/json"
;
run;
```

By invoking DIFY platform APIs through SAS, we implement comprehensive database management functionalities including:

1. initializing new database instances;
2. retrieving existing text segments to prevent redundant uploads;
3. inserting novel text segments;
4. querying the database for the top N most semantically similar segments to input text; and
5. deleting database instances post-use.

This operational framework enables robust database management and interaction.

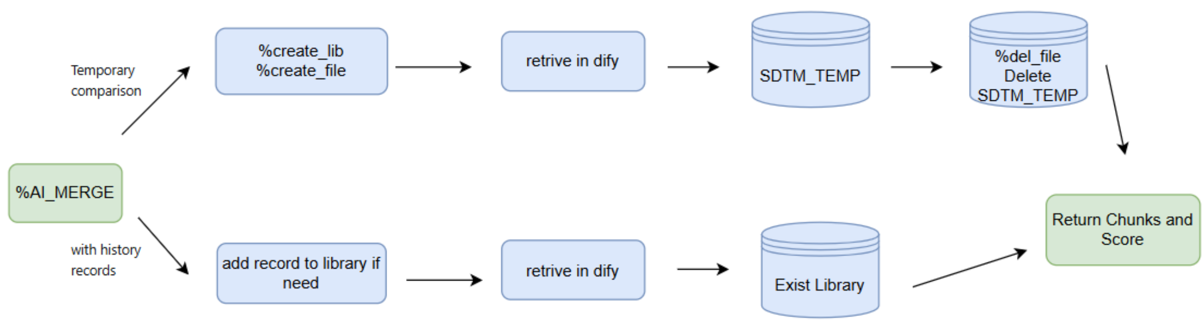

Figure 3 different ways to calculate similarity

KEY FEATURES

- Simple Operation Method for using the EMBEDING model in SAS.
- DIFY is characterized by its intuitive user interface and robust support for interface-based operations, consequently streamlining the maintenance of databases and eliminating procedural obstacles.

CONCLUSION

We have evaluated the application of current AI technologies in SDTM mapping. Two primary implementation strategies emerge: a comprehensive end-to-end Python system leveraging embedding and chat models, or integrating these capabilities into existing SDTM workflows. The latter uses DIFY's database API with SAS proc HTTP to enable semantic matching, enhancing field search, retrieval, and cross-referencing. Given SDTM's inherent structural nature—which offers greater automation suitability than ADAM or TFL—we contend that adopting these techniques will enhance procedural logic while overcoming conventional programming constraints.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Huan Song
BeiGene, Inc
huan.song@beigene.com

Jie Liu
BeiGene, Inc
jie1.liu@beigene.com

Chunpeng Zhao
BeiGene, Inc
chunpeng.zhao@beigene.com

Guangan Cao
BeiGene, Inc
guangan.cao@beigene.com