Paper ML09

Accelerating Data Integration: Utilizing Python Scripts and Semantic Search for Variable Mapping in Multiple Studies

Xiaopeng Li, Merck & Co., Inc., Rahway, NJ, USA

ABSTRACT

Integrating data analysis often involves mapping ADaM variables across multiple studies. This process can be time-consuming for statistical programmers due to 1) variations in variable names/labels in ADaM Datasets and 2) variations in the organization of information in SDTM domains across multiple studies. This paper explores using Python scripting and semantic search to expedite variable mapping. The paper presents a case study of integrating data from 44 studies/protocols, with the goal of mapping a list of target variables. For each target variable, the semantic search recommends the most similar candidate variables from the source datasets, by using an open-source, pre-trained language model. Statistical programmers can then review the recommendations and refine the variable mapping based on their domain knowledge and standard practices. The semantic search is facilitated by the SentenceTransformers framework from the Hugging Face platform, a popular open-source resource.

1. INTRODUCTION

The purpose of this paper is to explore the innovative use of semantic search to speed up the variable mapping process. It starts with an introduction to the key concepts and components of semantic search, such as embedding spaces, Sentence Transformers, and Vector Databases. The focus then shifts to how these technologies are applied in variable mapping and the considerations during their application. Finally, the results and implications are discussed. The goal of this paper is to present a proof of concept and to explore the feasibility of applying semantic search for data integration. The authors do not claim that the proposed approach is currently incorporated into any standard processes.

2. SEMANTIC SEARCH AND RELATED CONCEPTS

Semantic search is a powerful technique that leverages advanced natural language processing (NLP) models to understand the meaning and context of words and phrases, rather than just matching keywords. Central to semantic search is the idea of an embedding space, a high-dimensional area where data objects like text, videos, and audio files are represented as vectors. These vectors capture the semantic relationships between different data points, allowing similar concepts to be located near each other [1].

Sentence Transformers are a type of model that further enhances the capabilities of semantic search. Introduced by the Transformer model architecture, Sentence Transformers can generate fixed-size vector representations (embeddings) of sentences, capturing their semantic meaning. This is particularly useful for tasks such as semantic similarity, information retrieval, and clustering [2].

To efficiently manage and query these embedding vectors, vector databases come into play. Vector databases store the vector representations and facilitate fast similarity searches. One popular example is FAISS (Facebook AI Similarity Search), which is designed to handle large-scale vector data. By using vector databases, semantic search systems can quickly identify and retrieve the most relevant data points based on their semantic similarity, significantly improving the accuracy and speed of information retrieval [3].

The process of semantic search begins with creating embedding vectors for both the query and the source texts. Next, the distances between these vectors are calculated, and the source text with the shortest distance to the query is identified as having the most similar semantic meaning.

3. APPLICATION IN VARIABLE MAPPING

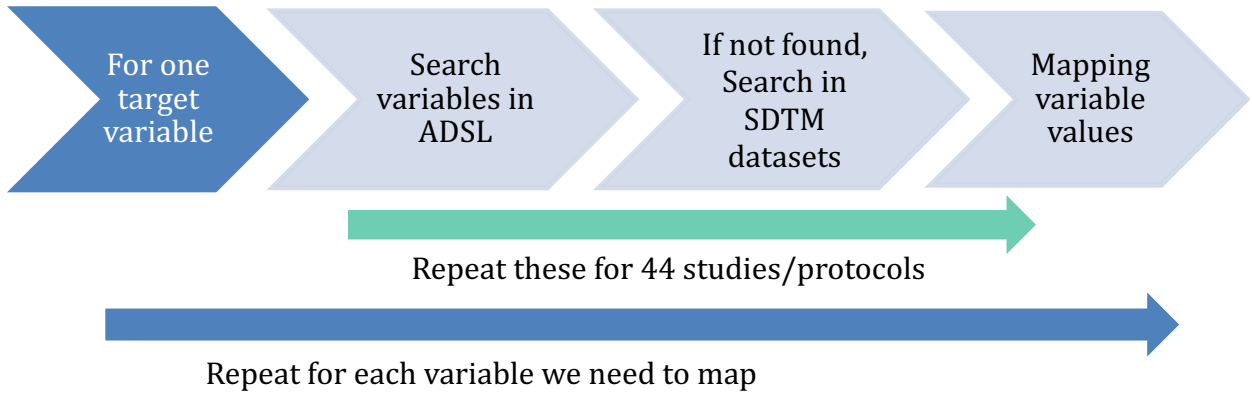
3.1 CASE STUDY

Cumulative safety datasets (CSD) are safety datasets which are integrated from 44 studies and protocols from a single clinical compound. These studies and protocols were not integrated into the pool all at once; rather, they were added incrementally over the years.

The objective of this case study project is to map a set of additional ADSL variables, either from the ADSL of individual studies or their SDTM datasets, simultaneously from 44 studies and protocols.

The graph below demonstrates the traditional process for completing this project: For each target variable we wish to map, we first examine one source study/protocol's ADSL. If the desired ADSL variable is not found, we then check the study's SDTM datasets. This procedure needs to be repeated 44 times to complete the mapping for one target variable.

Figure 1. Variable mapping process



3.2 CHALLENGES

1) The process becomes increasingly time-consuming as large number of studies/protocols are involved. There are large number of ADSL variables and an even larger number of SDTM datasets and variables to search through. On average, each ADSL across the 44 studies/protocols contains 115 variables, with the number ranging from 68 to 161. On average, each study/ protocol has 36.8 SDTM datasets.
2) Variations in ADSL variable names and labels present significant challenges for the task. As illustrated in the tables below, we often encounter discrepancies between the source and target variables' names and labels, although they are the same by semantic meanings. This issue is inherent to all data integration projects, regardless of the number of studies involved.

Table 1. Variations in ADSL variable name and labels

	Variable Name	**Variable Label**
Target variable	DCTREAS	Reason for Discontinuation of Treatment
Source Study #1 ADSL	STATUS	Reason for Discon. From Study Treatment
Source Study #2 ADSL	No variable found in ADSL. Need to derive from SDTM.DS	
Source Study #3 ADSL	DSTATUSM	Reason for Discon. From Study Treatment
Source Study #4 ADSL	TRDCREAS	Reason for Treatment Discontinuation

3) When we need to map a variable from SDTM datasets, we typically know which SDTM domain contains this information. For example, "Reason for discontinuation" is usually found in the Disposition domain, and "Baseline weight" is collected in the Vital Sign domain. However, variations in the organization of information across different studies and protocols can present challenges. This makes it challenging and time-consuming to identify which SDTM domain dataset contains the relevant information. For instance, while DTHCAUS (Cause of Death) is usually recorded in the “Death Details” domain, it may be collected in the Clinical Events domain for older studies which started before the release of SDTM IG 3.2.
4) Additionally, there are variations in code lists and variable values.

3.3 PROPOSED SOLUTIONS

It becomes clear that semantic search could offer a solution to the challenges presented by the variations in ADSL variable labels.
When we try to apply semantic search to assist with SDTM mapping issues, however, another challenge arises. SDTM datasets are structured, whereas Sentence Transformers and Large Language Models (LLMs) are particularly well-suited for handling unstructured data. How do we address this issue? The proposed solution is to create “sentences” to represent key topics collected in SDTM datasets.
Let’s step back and review SDTM domains and variables. The three general observation classes in SDTM datasets are Interventions, Events, and Findings. Additionally, there are domain datasets that include subject-level data not fitting into the general observation classes, such as Demographics (DM), Comments (CO), Subject Elements (SE), and Subject Visits (SV). Trial Design Model (TDM) datasets like Trial Arms (TA) and Trial Elements (TE) [4]. Relationship datasets, such as RELREC and SUPP, connect information from different datasets. When examining the target variables we want to map, it becomes evident that we only need to focus on SDTM datasets categorized under Interventions, Events, and Findings. For simplicity, we also exclude the SUPP domain.

SDTM variables are categorized into five main roles: Identifier variables, Topic variables, Timing variables, Qualifier variables, and Rule variables. Upon examining the example dataset in Table 2, we observe that qualifier and topic variables represent the topics collected in this dataset. We are not interested in identifier, timing and rule variables.

Table 2. Sample SDTM.MH dataset

Identifier	Identifier	Qualifier	Topic	Timing
SUBJID	MHSEQ	MHCAT	MHTERM	VISIT
202401	1	PRIMARY DIAGNOSIS	MELANOMA	Visit 1
202402	2	PRIMARY DIAGNOSIS	hyperlipidemia	Visit 1
202403	3	PRIMARY DIAGNOSIS	anxiety states	Visit 1

The SDTM “sentences” are created by utilizing dataset’s label, the qualifier and topic variables for SDTM datasets under Interventions, Events, and Findings categories. Here are two examples of unstructured SDTM “sentences” which represent the key topics collected in structured SDTM dataset DS and TR:

1) dataset is Disposition, DSCAT is DISPOSITION EVENT, DSTERM is ADVERSE EVENT, PROGRESSION OF DISEASE UNDER INVESTIGATION, NEVER ENTERED FOLLOW UP, DID NOT MEET PROTOCOL ELIGIBILITY, SUBJECT WITHDREW CONSENT, LOST TO FOLLOW-UP, ADMINISTRATIVE, SUBJECT DID NOT WISH TO CONTINUE FOR REASONS UNRELATED TO ASSIGNED STUDY TREATMENT, NON-COMPLIANCE WITH PROTOCOL, INVESTIGATOR DEEMS IT IS NOT IN SUBJECT'S BEST INTEREST TO CONTINUE
2) dataset is Tumor Results, TRCAT is , TRTEST is Tumor State, unique values are PRESENT, , ENLARGEMENT, NOT EVALUABLE, ABSENT

3.4 THECHNICAL IMPLEMENTATIONS

3.4.1 Embedding Model

An embedding model is chosen to create embeddings. The BioLORD-2023 model [5], a state-of-the-art tool designed for semantic textual similarity (STS) and biomedical concept representation (BCR) within the clinical domain, is selected. To avoid API calls from the Hugging Face website, the trained model weights are downloaded to the local drive and utilized from local drive for project implementation.

3.4.2 Vector Database

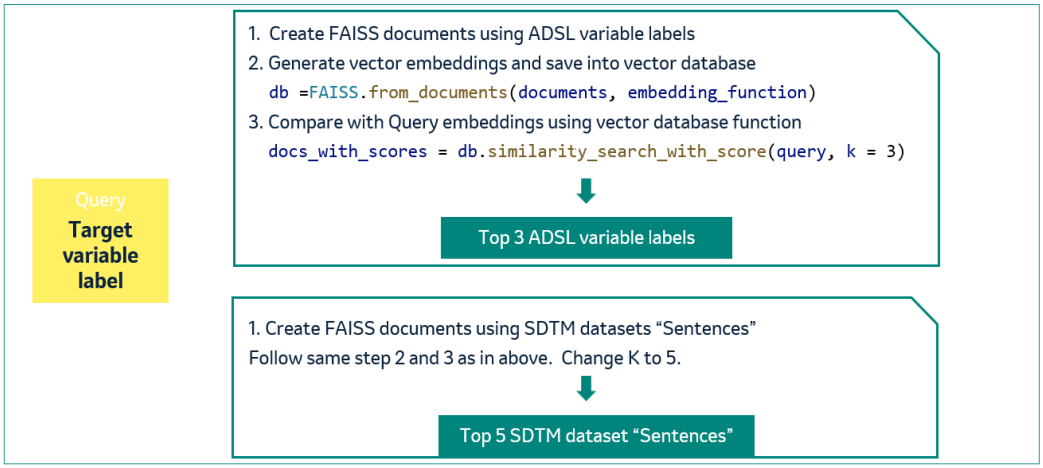
The FAISS library from LangChain is employed to create documents, store embedding vectors, and implement semantic search.

3.4.3 Implementation plan

The target variable label is used as a query. When mapping ADSL variables, all the ADSL variable labels in the source studies/protocols are converted into FAISS documents. These documents, along with the embedding model, are integrated into the FAISS instance to create a FAISS vector database. Subsequently, the vector database’s similarity search function is employed to locate the top three most similar ADSL variables.

When mapping from SDTM datasets, we transform SDTM 'sentences' into FAISS documents and follow the same procedures. The only distinction is that we select the top 5 'sentences' that are most relevant to the query.

Figure 2. Implementation plan



4. RESULTS

The results are saved into an excel sheets with two tabs. The first tab has the “recommended” top 3 ADSL variables for each target variable and for each source ADSL (see Table 3). The second tab has the top 5 “recommended” SDTM “sentences” (see Table 4). The programmers can review these “recommended” variables and SDTM source datasets and make the final mapping decisions. The unique values column in Table 3 are not included in semantic search but is included in the outputs to facilitate the variable value derivation across studies.

Table 3. Mapping recommendations from ADSL as source

Target var	Target label	Source dataset	Source Var	Source label	Unique values *
DCTREAS	Reason for Discontinuation of Treatment	lpt_001	STATUS	Reason for Discon. From Study Treatment	['ADVERSE EVENT' 'PROGRESSIVE DISEASE'
DCTREAS	Reason for Discontinuation of Treatment	lpt_001	DISCNDTC	Date/Time of Discontinuance	['2013-07-23' '2014-02-13'
DCTREAS	Reason for Discontinuation of Treatment	lpt_001	DAYOFDC	Relative Day of Discontinuance	[105. 238. 20. nan 365.
DCTREAS	Reason for Discontinuation of Treatment	lpt_002	DAYOFDC	Relative Day of Discontinuance	[186. 275. nan 21. 227.
DCTREAS	Reason for Discontinuation of Treatment	lpt_002	RFENDTC	Subject Reference End Date/Time	['2013-07-29T13:00' '2013-11-26T14:28'
DCTREAS	Reason for Discontinuation of Treatment	lpt_002	RFENDT	Subject Reference End Date	[datetime.date(2013, 7, 29) datetime.date(2013, 11, 26)

*The unique values are truncated for fitting in this paper.

Table 4. Mapping recommendations from SDTM as source

Target var	Target label	Source Study	Sentences*
DCTREAS	Reason for Discontinuation of Treatment	src_002	dataset is Disposition, DSCAT is DISPOSITION EVENT, DSTERM is LOST TO FOLLOW-UP, PROGRESSIVE DISEASE,..
DCTREAS	Reason for Discontinuation of Treatment	src_002	dataset is Clinical Findings, CFCAT is SURVIVAL FOLLOW-UP, CFTEST is Survival Status, unique values are ALIVE, DEAD, SUBJECT WITHDREW CONSENT, …
DCTREAS	Reason for Discontinuation of Treatment	src_002	dataset is Clinical Findings, CFGRPID is SFU, CFTEST is Survival Status, unique values are ALIVE, DEAD, SUBJECT WITHDREW CONSENT, …
DCTREAS	Reason for Discontinuation of Treatment	src_002	dataset is Intervention Findings, IFCAT is OTHER SUSPECT THERAPY, IFTEST is Action Taken, unique values are DRUG WITHDRAWN, DOSE NOT CHANGED,…
DCTREAS	Reason for Discontinuation of Treatment	src_002	dataset is Disposition, DSGRPID is DS02, DSTERM is PROGRESSIVE DISEASE, WITHDRAWAL BY SUBJECT, ADVERSE EVENT, ….

*The unique values are truncated for fitting in this paper.

4.1 EVALUATING ACCURACY

Considering that the results function as recommendations from the semantic search, we employed a recommendation system metric known as MRR (Mean Reciprocal Rank) [6] to evaluate the accuracy of our mapping results from ADSLs.

MRR, or Mean Reciprocal Rank, measures the average of the reciprocal ranks of results for a series of queries, providing insight into how effectively the system ranks relevant items at the top. An MRR score closer to 1 indicates excellent search quality. Figure 3 illustrates the calculation of Reciprocal Rank:

In cases where the correct ADSL variable is not within the top 3 results, we assign Reciprocal Rank (RR) scores based on two scenarios:

Scenario 1: If a correct source ADSL variable exists but is missing from the top 3 list, the RR score is set to 0.

Scenario 2: If the related information is genuinely absent from the source ADSL, the RR score is set to 1.

Based on the these rules, we calculate the Reciprocal Rank for the results. Table 3 illustrates two successful cases with RR equal to 1: the first case where the correct variable is placed in the first position, and the second case where relevant information is absent from ADSL.

Figure 3. Reciprocal Rank calculation

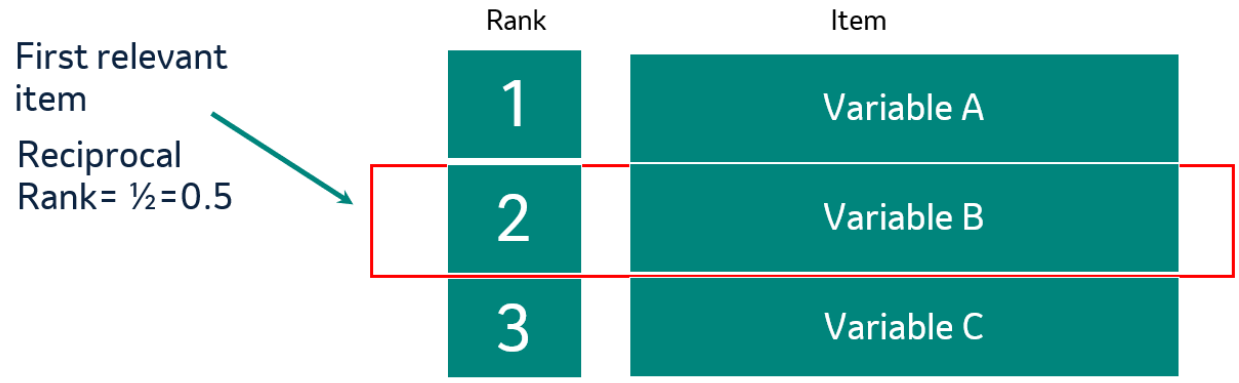


Table 5 illustrates an unsuccessful case where a correct ADSL variable exists, but the method fails to place it within the top three recommendations. The RR for this case is 0.

Table 5. One unsuccessful case

Target Var	Target Label	Source Var	Source Label	Correct Var	Correct Label
DCTREAS	Reason for Discontinuation of Treatment	DCSREAS	Reason for Discontinuation from Study	DCTREAS1	Reason for Discon of Treatment P1
		DCTREAS2	Reason for Discon of Treatment P2		
		EOTSTT1	End of Treatment Status Part 1		

The Mean Reciprocal Rank (MRR) for all queries is 0.97. This indicates that in nearly every instance, the correct matching variable is consistently placed at the highest rank. Or, when there is no correct variable within the top 3 list, it is true that the information is absent from the ADSL dataset.
For queries where variable name/label not the exact same, the MRR is impressively high at 0.82. This indicates that, in a significant majority of these cases, the appropriate matching variable is ranked either first or second.
Overall, these metrics demonstrate the robustness and efficiency of the semantic search algorithm in retrieving pertinent information, even in circumstances where exact matching is lacking.

5. DISSCUSION

Semantic search plays a crucial role in the initial stages of data integration by generating recommendations for variable and dataset selection. However, it is essential to recognize that the final judgment rests with statistical programmers who must review these recommendations carefully. Leveraging their domain expertise and adhering to their company's standard operating procedures (SOPs), these professionals ensure the appropriateness and relevance of selected variables and datasets. This human-in-the-loop approach is vital, as it grounds the decision-making process in practical experience. By integrating human judgment into the technology-driven recommendations of semantic search, the quality and accuracy of data integration can be ensured.

On the technology front, recent advancements in semantic search, including embedding models, vector databases, and sophisticated search algorithms, have resulted in a wealth of ready-to-use products that simplify the analysis process. Nevertheless, the successful application of these tools, particularly in the context of variable mapping, relies heavily on the domain knowledge and experience of statistical programmers. Their expertise in critical areas such as Study Data Tabulation Model (SDTM), Analysis Data Model (ADaM), and the broader data integration process is critical. Without this specialized knowledge, the potential of semantic search technologies may not be fully realized, and the effectiveness of data integration efforts could be compromised. As such, a collaborative approach that combines advanced technology with skilled human oversight is essential for optimizing data analysis in complex environments.

6. CONCLUSION

Semantic search offers statistical programmers a systematic and reliable method for accurately locating source

variables and datasets. This is particularly useful for data integration tasks involving either a large number of studies or partners' clinical data, where complexity is increased due to varying terminologies and data structures. By using semantic search, the time needed to identify the correct variables and datasets is significantly reduced. As a result, data integration projects can get a jump start, allowing the project team to identify potential issues earlier. This early identification fosters discussion and the development of solutions to challenges, ultimately improving overall project efficiency.

REFERENCES

1. Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. International Conference on Learning Representations.
2. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 3980-3990.
3. Facebook AI. (2020). FAISS: A library for efficient similarity search and clustering of dense vectors.
4. A Guide to CDISC SDTM Standards and Domains. https://www.quanticate.com/blog/bid/51830/cdisc-sdtm-v3-1-2-theory-and-application
5. François Remy, Kris Demuynck, Thomas Demeester (2024). BioLORD-2023: semantic textual representations fusing large language models and clinical knowledge graph insights. Journal of the American Medical Informatics Association, Volume 31, Issue 9. https://doi.org/10.1093/jamia/
6. Voorhees, E. M. (1999). The TREC-8 question answering track report. In Proceedings of the Eighth Text Retrieval Conference (TREC-8).

ACKNOWLEDGMENTS

The authors would like to thank BARDS Statistical Programming Leadership and Management at Merck & Co., Inc., Rahway, NJ, USA. The authors also greatly appreciate the Merck oncology ISS and CSS team.

CONTACT INFORMATION

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

Xiaopeng Li
Associate Principal Scientist
Biostatistics and Research Decision Sciences
Oncology Statistical Programming
Merck & Co., Inc., Rahway, NJ, USA
xiaopeng.li@merck.com