Paper ET19

Balance between Human and Machine
- How NLP helps translate text specification to code, and vice versa

Hao Chen, Yiwen Wang, AstraZeneca China Biometrics, Shanghai, China
Jaimin Patel, AstraZeneca US Biometrics, Delaware, US

ABSTRACT

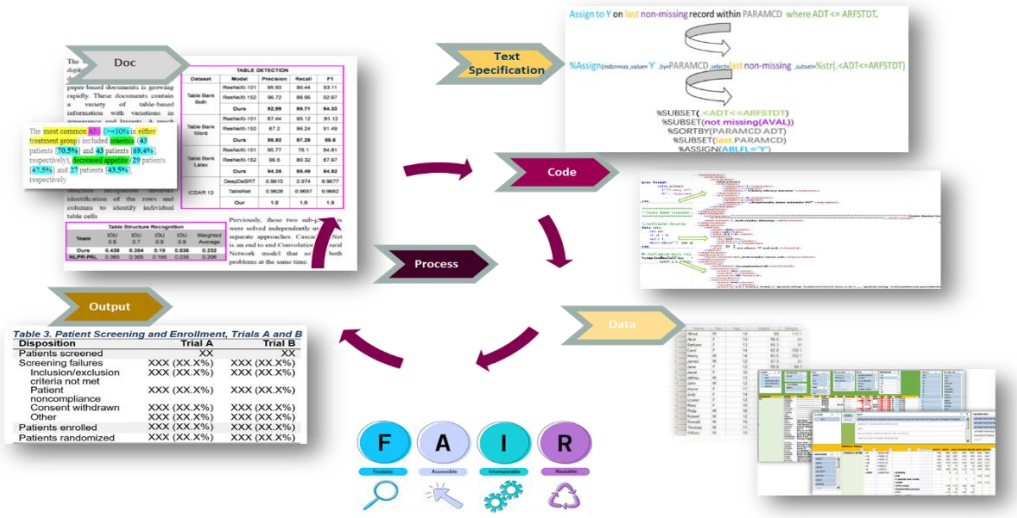
Clinical data analysis algorithms are complex and usually pre-planned in human-readable text specification before execution by machine-readable programing codes. Programmers spend lots of efforts on creating the text specifications and translating them into executable codes, and this work is not one-time effort with the change from source data and code debugging process, so keeping synchronization of the specifications and code is also not an easy work. This paper introduces how natural language processing (NLP) can help us solve this problem, utilizing the natural language understanding (NLU) to help translate from text specifications to codes, and natural language generation (NLG) to translate from codes back to text specifications when needed. This approach will be demonstrated with an Rshiny app for SDTM mapping process. We also explored the ChatGPT solution and discussed about the potential of NLP for other Biometrics tasks in this framework.

INTRODUCTION

The current data flow of traditional clinical trials involves a convoluted and non-linear process that requires a great deal of manual effort and often leads to rework and inefficiencies. However, the future state of clinical trials involves a more streamlined and interoperable approach. This will be made possible through the standardization of data and vocabulary, as well as through leveraging the insights gained from AI and machine learning analysis. By embracing these advancements, the clinical trial process can become more efficient and effective, ultimately leading to better outcomes for patients.

In clinical research, digitalization of trial artifacts, including documentation, programming codes, data, output tables, and complex processes, is critical for achieving FAIRness (Findable, Accessible, Interoperable, and Reusable). By converting these artifacts into metadata and systematically managing them, clinical data analysis can be conducted more efficiently, leading to better decision-making.

Natural Language Processing (NLP) is a field of AI that aims to bridge the gap between human communication and computer understanding. It is essential for processing large amounts of textual data and resolving ambiguity in language to structure unstructured data sources. NLP has two major components: Natural language understanding (NLU) and Natural language generation (NLG). NLU helps computers understand human language by parsing text and extracting meaning, while NLG generates human-like text. This technology can be used for various applications, such as translating text to code and vice versa. Both rule-based systems and machine learning can be used to implement NLU and NLG, depending on the complexity of the language task. Rule-based systems are effective for highly structured and constrained language, while machine learning is preferred for more complex and varied language tasks.



BUSINESS PROBLEM

Developing clinical data analysis algorithms is a complex process that typically requires pre-planning in human-readable text. However, strictly adhering to standard algorithm descriptions can be challenging for human users. While Graphical User Interfaces (GUIs) can provide an alternative means of building flexible algorithms, they are often difficult to use. As a result, algorithms written in free text are commonly used in the industry, leading to diverse mapping specifications across therapeutic areas and companies. Translating human-readable text to machine-readable code is a time-consuming process, and ongoing updates are necessary to accommodate changes in source data and the debugging process. In daily work, synchronizing specifications and code can present a challenging task for human users.

Define-XML Version 2 [Variable Level Metadata]

Demographics (DM) [Location: dm.xpt]

Variable	Label	Key	Type	Length	Controlled Terms or Format	Origin	Derivation/Comment
STUDYID	Study Identifier	1	text	40		Protocol	
DOMAIN	Domain Abbreviation		text	2		Assigned	Two-character abbreviation for the domain most relevant to the observation. The Domain abbreviation is also used as a prefix for variables to ensure uniqueness when datasets are merged.
USUBJID	Unique Subject Identifier	2	text	20		Derived	Concatenation of STUDYID and SUBJID
SUBJID	Subject Identifier for the Study		text	3		CRF Page 1	
RFSTDTC	Subject Reference Start Date/Time		datetime		ISO8601	Derived	RFSTDTC = First date/time of study drug treatment
RFENDTC	Subject Reference End Date/Time		datetime		ISO8601	Derived	RFENDTC = Date of Completion or date of Withdrawal. Null for screen failures
RFXSTDTC	Date/Time of First Study Treatment		datetime		ISO8601	Derived	RFXSTDTC = Date/time of first exposure to protocol specified treatment from CRF
RFXENDTC	Date/Time of Last Study Treatment		datetime		ISO8601	Derived	RFXENDTC = Date/time of last exposure to protocol specified treatment from CRF

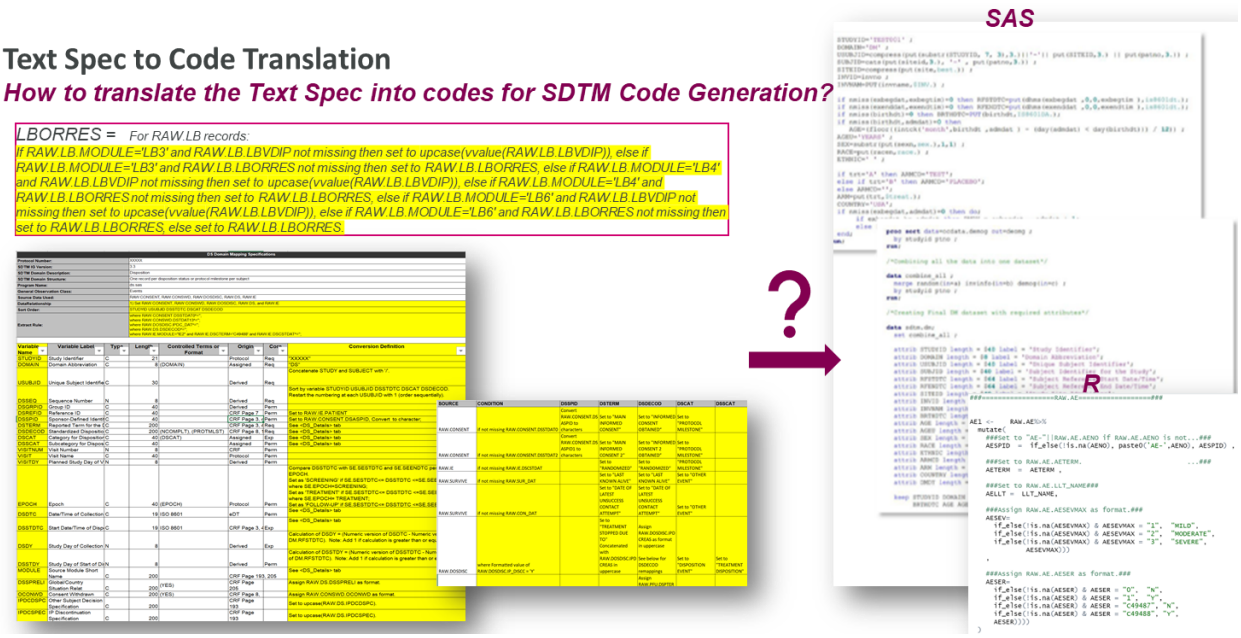
To address these challenges, leveraging AI and NLP technologies can make a significant difference in the clinical data analysis process. These technologies can help automate the translation of human-readable text to machine-readable code, reducing the time required for updates and synchronization. Additionally, AI and NLP can help enhance the flexibility and user-friendliness of algorithms, resulting in more effective clinical data analysis.

CASE SHARING

As a case sharing exercise, we present a proposed solution for the SDTM Mapping case, which involves translating Text Specifications to Code using AI and NLP technologies. Our proposed solution includes SAS/R code generation, and algorithm text generation. By automating the translation of Text Specifications to Code, our solution can significantly reduce the time required for updates and synchronization, while also enhancing the user-friendliness and flexibility of algorithms. Through this case sharing exercise, we aim to demonstrate the potential of AI and NLP technologies in facilitating the translation of human-readable text to machine-readable code, revolutionizing clinical data analysis, and enabling more effective and efficient decision-making in the field.

OBJECT

The SDTM mapping file is typically composed of multiple sheets in an Excel file, which contain mapping algorithms at different levels, including data, variable, and value levels. Meanwhile, the target SAS/R codes consist of several sections, such as the header, macro variable definition, variable attributes definition, data join, and variable and value level derivation. To facilitate the generation of SDTM data from the Excel mapping spec, an automated process is designed to analyze the Excel file and produce the corresponding SAS/R code for execution.



PROCESS

In the context of Text to Code generation, the following five steps can be identified:

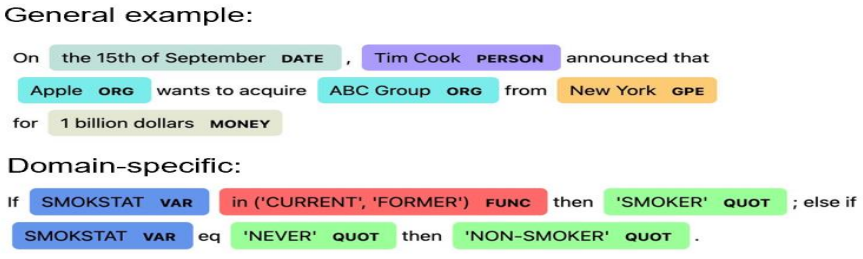
1. Pre-processing: In some cases, long texts may need to be split into shorter sentences to facilitate subsequent processing.
2. Named Entity Recognition (NER): Variables and formulas can be masked using NER to identify their presence and position within the text.
3. Normalization: The text can be semantically matched to a standard algorithm using embedded vectors to ensure consistency and facilitate parsing.
4. Parsing: The normalized algorithm can be parsed into an Abstract Syntax Tree (AST) for more structured processing.
5. Translation: Finally, the AST can be translated into code that can be executed by a computer.

When generating Text from Code, the following two steps can be followed:

1. Code Parsing: The input code can be parsed into an Abstract Syntax Tree (AST) for more structured processing.
2. Translation: The AST can be translated into text, or into other programming languages like R, Python or SQL if needed.

In the above steps, Named Entity Recognition (NER) and Normalization rely on machine learning techniques, while the other steps are based on rule-based algorithms.

NER (Named Entity Recognition)



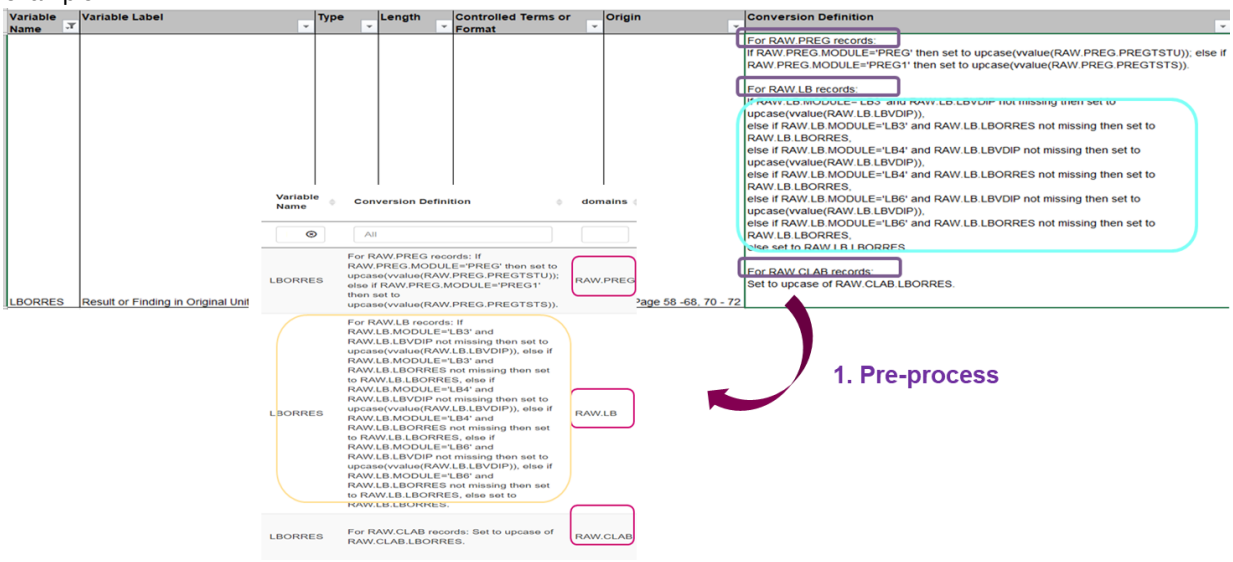
Normalization (Spec smart match)

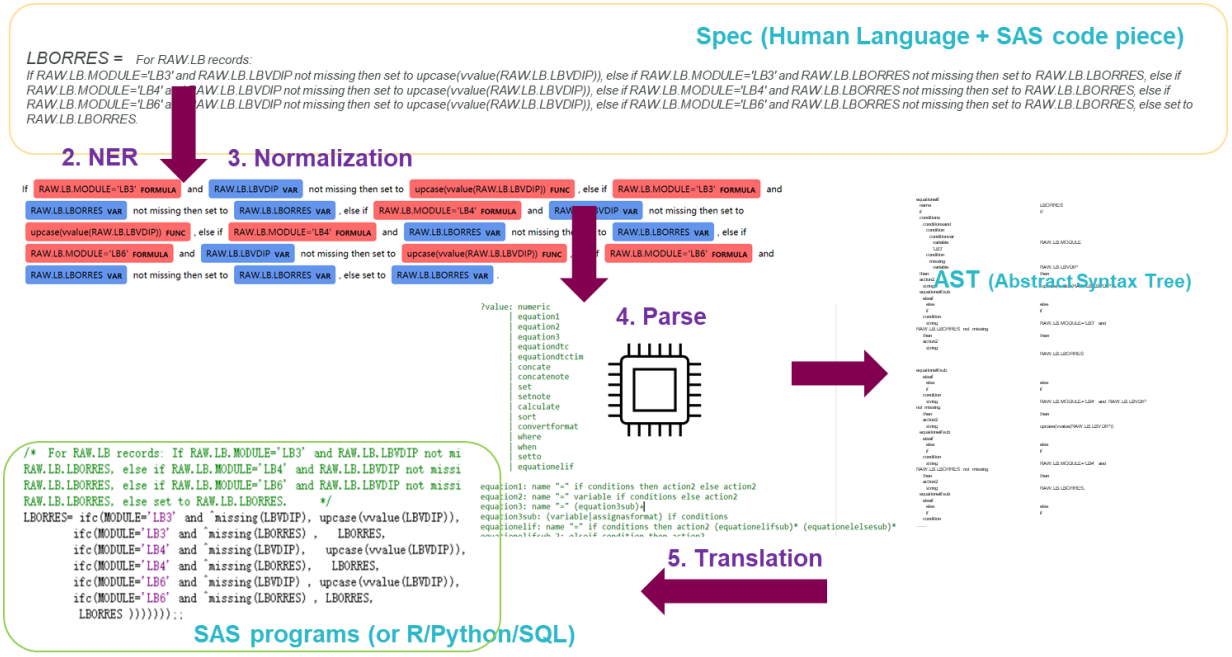
Flexible Spec	Variable masked	Standard Spec

	inputSpec	Input Spec normalized	Standard Spec matched	Similarity
0	Baseline flag	BASELINE FLAG	DERIVE BASELINE FLAG	90.0%
1	Convert date (VISDAT/VSDAT) to ISO8601 date.	CONVERT DATE ($$/$$) TO ISO8601 DATE	CONVERT DATE ($$/$$) TO ISO8601 DATE FORMAT	95.0%
2	Set to RAW.AE.AENO	SET TO $$	SET TO $$	100.0%
3	Convert To Char	CONVERT TO CHAR	CONVERT TO CHARACTER VERSION	90.0%
4	Convert To $AESEV. Format	CONVERT TO $$$. FORMAT	CONVERT TO $XX. FORMAT	95.0%
5	Sort by variable STUDYID USUBJID AESTDTC AEDECOD. Restart the numbering at each USUBJID with 1	SORT BY VARIABLE $$ $$ $$ $$. RESTART THE NUMBERING AT EACH $$ WITH 1	SORT BY VARIABLE $$ $$ $$. RESTART THE NUMBERING AT EACH $$ WITH 1 (ORDER SEQUENTIALLY)	95.0%
6	Concatenate (RAW.AE.AESTDAT and RAW.AE.AESTTIM) by '/'	CONCATENATE ($$ AND $$) BY @@	CONCATENATE $$, $$ BY @@	95.0%
7	Calculation of AESTDY = (Number version of date of AESTDTC - Numeric version of date of DM.RFSTDTC). Note1: Add 1 if greater than or equal to 0.	CALCULATION OF $$ = (NUMBER VERSION OF DATE OF $$ - NUMERIC VERSION OF DATE OF $$). NOTE1: ADD 1 IF GREATER THAN OR EQUAL TO 0.	CALCULATION OF $$ = (NUMERIC VERSION OF $$ - NUMERIC VERSION OF $$). NOTE: ADD 1 IF CALCULATION IS GREATER THAN OR EQUAL TO 0	89.6%

SOLUTION

Here is the process of how we convert the text algorithms into SAS code, taking variable LBORRES derivation for example:





Achieving full automation in the process can be challenging, and it is important to involve humans in the review and decision-making process for new scenarios, while also adding new patterns to the knowledge base as needed. For example, if the derivation for a variable such as AEACN is unrecognizable and cannot be parsed into SAS codes, the system will automatically flag this new scenario, requiring a programmer to write ad hoc open code to handle it. If this pattern occurs frequently, we can add it to the system library for future re-use.

Variable	Conversion Definition	sas	flag
STUDYID	" 00008"	STUDYID=" 00008"	Y
USUBJID	Concatenate STUDY and SUBJECT separating by '/'. (e.g., 00008/ 001).	USUBJID=cat(strip(STUDY),'/', strip(SUBJECT))	Y
AESEQ	Sort by variable STUDYID USUBJID AESTDTC AEDECOD. Restart the numbering at each	%sdtm_seq(indata=&syslast,domain=&sdtm_domain.);	Y
AESPID	Set to "AE-"||left(trim(put(RAW.AE.AENO,best.))) if RAW.AE.AENO is not missing.	if ^ missing(AENO) then do; AESPID = "AE-"||left(trim(put(AENO,best.))); end;	Y
AETERM	Set to RAW.AE.AETERM.	*AETERM=AETERM; *copy/keep;	Y
AELLT	Set to RAW.AE.LLT_NAME	AELLT=LLT_NAME	Y
AECAT	Assgin RAW.AE.AECAT as format If RAW.AE.AECATOCC=1	if AECATOCC=1 then do; AECAT = ifc(AECAT = "1", "ANAPHYLAXIS AND OTHER SERIOUS HYPERSENSITIVITY REACTIONS", ifc(AECAT = "2", "INFUSION-RELATED REACTIONS"	Y
AEACN	Set to RAW.AE.AEACN with below conversion.Note: 0 = DOSE NOT CHANGED, 3 = DRUG INTERRUPTED, 4 = DRUG WITHDRAWN, 98 = NOT APPLICABLE	AEACN= AEACN with below conversion. Note: 0 = DOSE NOT CHANGED, 3 = DRUG INTERRUPTED, 4 = DRUG WITHDRAWN, 98 = NOT APPLICABLE	N
AEREL	Assign RAW.AE.AEREL as format	AEREL =ifc(AEREL = "0", "NOT RELATED", ifc(AEREL = "1", "RELATED",	Y
AESCONG	When AESER = "Y": Assign RAW.SERAE.AESCONG as format.	if AESER="Y" then do; AESCONG = ifc(AESCONG = "0", "N", ifc(AESCONG = "1", "Y",ifc(AESCONG = "C49487", "N",ifc(AESCONG = "C49488", "Y", AESCONG)))) ;; end;	Y
AECONTRT	Set to substr (vvalue (RAW.AE.AETRT), 1,1)	AECONTRT=substr(vvalue(AETRT),1,1)	Y

It is crucial to note that when dealing with highly complex study-specific derivations, the chances of success for NLP analysis and code auto generation are limited, and the opportunity for reusability may also be limited. Therefore, in such cases, it may be necessary to manually perform the derivation using user-defined complex algorithms written in open code.

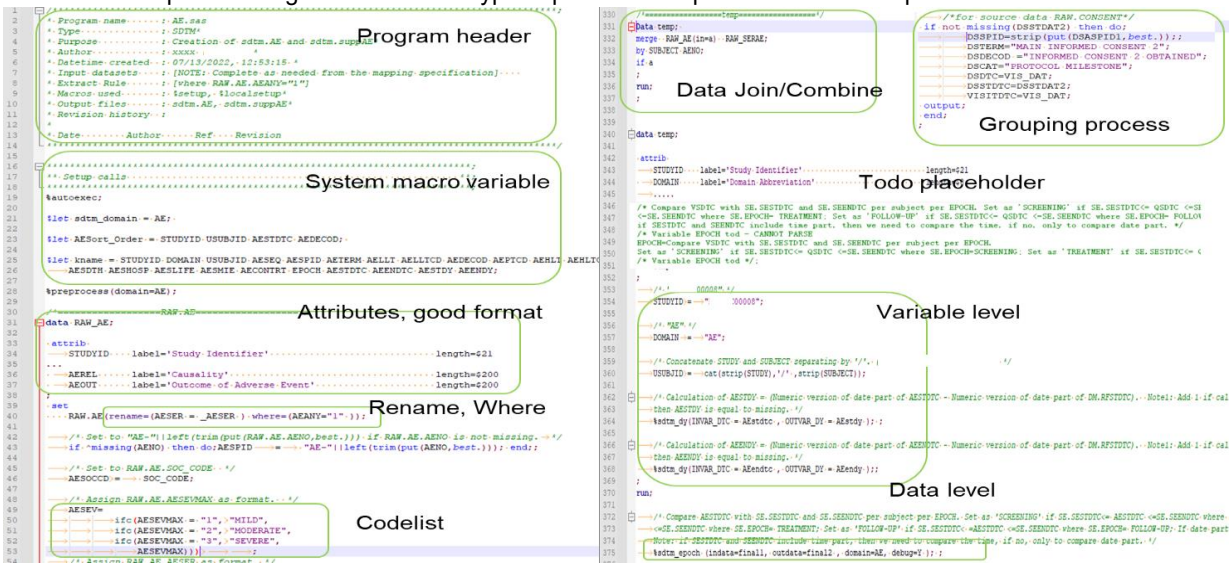
Taking all the scenarios into consideration, we divide the derivations into three types, including Type I (simple derivation like copy/drop/rename or function call), Type II (standard derivation involving macro calls like standard mapping or user-defined pattern), and Type III (complex derivation involving user-defined complex algorithms that typically requires manual coding). NLP-powered automation can be useful for Types I and II, while Type III is best left to human intelligence.

- **Type I (from Spec) – simple derivation**
  - **Copy/Drop/Rename**
  - **Assign - function call**
- **Type II (from Spec) – standard derivation (macro call)**
  - **Standard Mapping macro call (top down)**
  - **User-defined Pattern macro call, some chance to reuse (bottom up)**
- **Type III (from open code) – complex derivation**
  - **User-defined complex algorithm, little chance to reuse**
  - **Do it in open code manually**
  - **Rerun applicable with updated Code & Spec**

INTEGRATED PROCESS

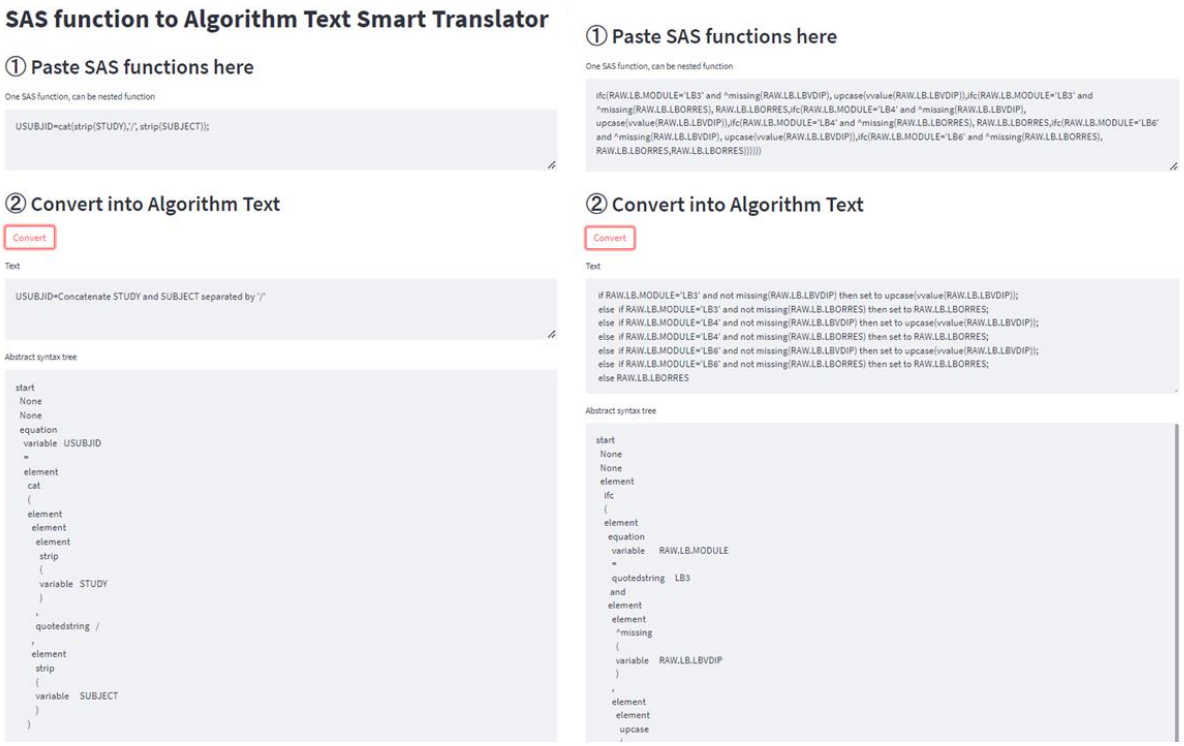
The R Shiny App is developed to provides a comprehensive solution for generating SAS code from an Excel Spec, allowing users to upload the file, select the target domain, and automatically generate the SAS code. If any Type III algorithms require manual coding, users can insert the open code and upload the revised code to the system. Additionally, the Excel Spec can also be updated if necessary. The final code is regenerated by combining the user's code with the system's automatically generated code. This process can be repeated multiple times if necessary, with Type I and II always coming from the Excel Spec and Type III being added by the user.

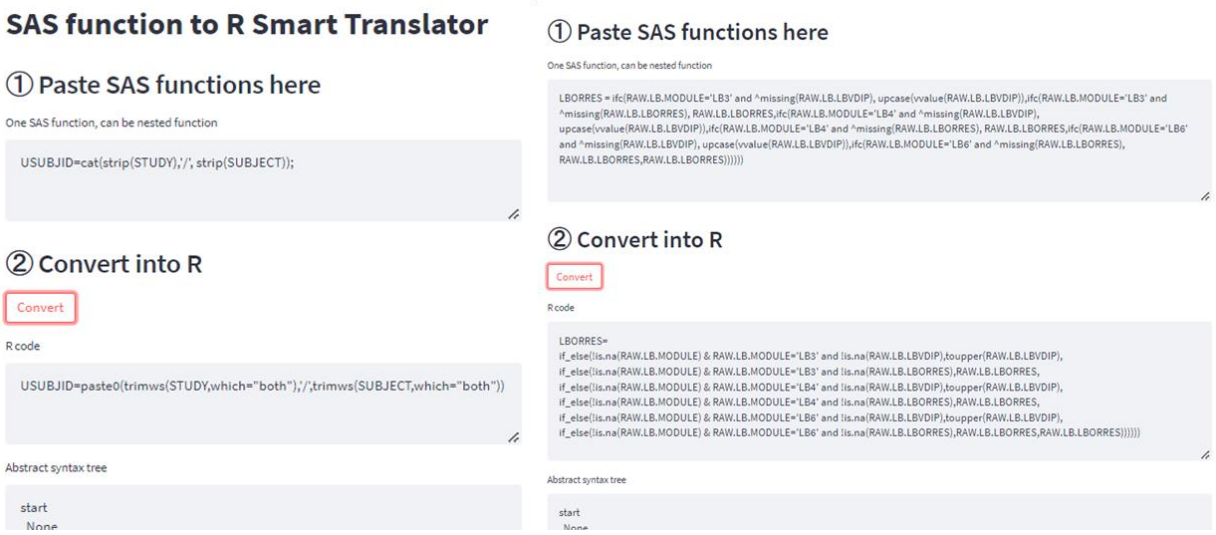
Here is an example of auto generated code. Type III part can be updated in the 'Todo placeholder'.



CODE TO ALGORITHM TEXT

As introduced above, when we have the code and want to get the algorithm text, it is easy to use rule-based system to parse the code into an Abstract Syntax Tree (AST) based on which can be translated into algorithm text, or into other programming languages like R, Python or SQL if needed.

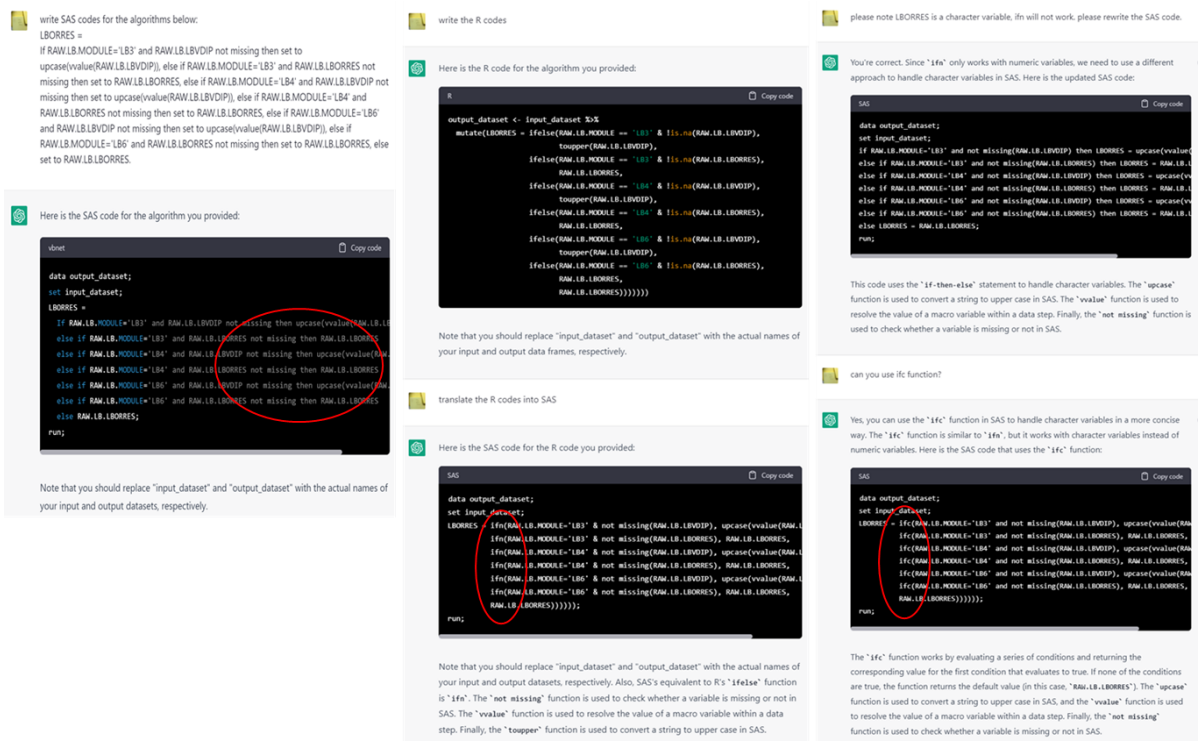




DISCUSSION

CHATGPT SOLUTION TESTING

As a powerful system excels in both NLP and code analysis & generation, ChatGPT has recently made many breakthroughs. So we did similar testing for the task at hand. We found that R code outperforms SAS code due to its access to more open-source training data, and translating code into R before translating it into SAS can improve the quality. Achieving accurate results may require multiple rounds of refinement and adjustment, while controlling the quality of its results can be challenging due to ChatGPT's black box nature. However, ChatGPT is capable of effectively processing very complex text, surpassing the capabilities of many self-developed systems. It's essential to consider the protection of company-restricted or confidential data when using ChatGPT.

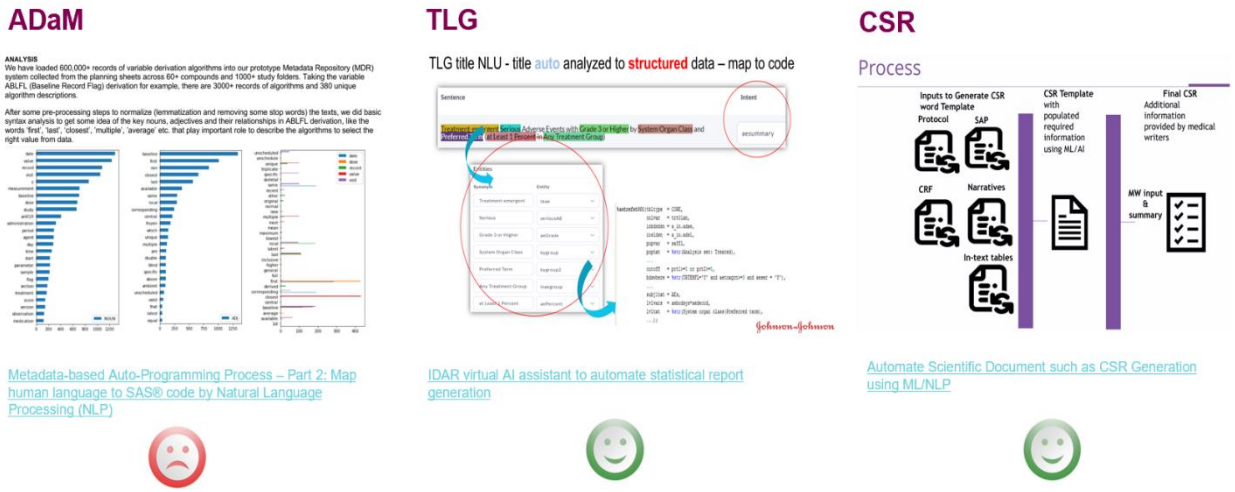


OTHER TASKS

In addition to its great benefits for the SDTM mapping process, the NLP solution can also be applied to other tasks such as ADaM, TLG, and CSR generation, provided that standard algorithm libraries and knowledge databases are available to support. Based on our observations of the progress in our industry's work, we would like to share our thoughts.

- ADaM: This task involves long, complex sentences and can be challenging for non-standard specifications. Standard macros/functions and knowledge databases are required for support.

- TLG: Short and relatively standard sentences are involved in this task, requiring support from standard macros/functions as well as table layout analysis from mockup tables.
- CSR: The NLP solution is effective in key information extraction, similarity calculation, and text summarization, and can predict and identify key information in CSRs and link it to different documents and sections.



SUMMARY

In this paper, we have introduced the use of natural language processing (NLP) to facilitate the translation of text specifications into code and vice versa. Our approach has been successfully demonstrated through an Rshiny app designed for the SDTM mapping process, and we have also explored the ChatGPT solution and discussed about the potential of NLP for other tasks in the biometrics process.

From our experience designing and testing the SDTM mapping process, we have observed the following benefits of the NLP solution and the combination of AI+HI:

- Natural language descriptions of algorithms reduce the requirements on specifications, making them applicable to different therapeutic areas and companies
- Code generation automation reduces the need for manual synchronization between specifications and code, increasing efficiency and reducing errors
- Cultivation of a user's mindset in standardization and automation helps to improve overall process consistency and efficiency
- Accumulation of an algorithm knowledge base over time can help improve the efficiency of future projects
- The framework is applicable for other programming languages beyond SAS, such as R, Python, and SQL
- NLP can benefit both upstream and downstream work, such as algorithms recommendation, dependency analysis, and Define.xml creation
- Machine learning can accelerate, simplify, and automate the tasks for conversion between unstructured and structured data, upon which rule-based systems can be built for automatic systems
- The combination of machine and human intelligence (AI+HI) revolutionizes the way we work, learn, and solve problems in the biometrics process.

It is crucial for our industry to remain up-to-date with the technological advances in AI and NLP, as their potential applications will only continue to grow. While these tools offer many potential benefits, it is important to be aware of the risks and limitations associated with their use. Proper validation and testing should be conducted to ensure the reliability and safety of these systems. By addressing these risks, we can ensure that the use of AI and NLP is both effective and responsible in our industry.

REFERENCES

1. AI/ML – Challenges and Opportunities in Implementing Analytics in the Pharma Industry. https://phuse.s3.eu-central-1.amazonaws.com/Archive/2021/SDE/APAC/Virtual%20-%20India%20Winter/PRE_India06.pdf

2. Automate Scientific Document such as CSR Generation using ML/NLP. https://www.lexjansen.com/phuse-us/2021/ml/PRE_ML01.pdf
3. ChatGPT: Optimizing Language Models for Dialogue. https://openai.com/blog/chatgpt
4. Metadata-based Auto-Programming Process. https://www.lexjansen.com/phuse-us/2018/si/SI10.pdf
5. Metadata-based Auto-Programming Process – Part 2: Map human language to SAS® code by Natural Language Processing (NLP). https://www.lexjansen.com/phuse-us/2019/ml/ML05.pdf
6. IDAR virtual AI assistant to automate statistical report generation. https://www.lexjansen.com/phuse-us/2020/ml/ML01_ppt.pdf

ACKNOWLEDGMENTS

We would like to express our gratitude to Meng Chen for her generous support in sponsoring this project. Special thanks to Kevin Huang, Xinya Wang, Zhanjian Dong, Kang Zhao, Fan Zhang, and Zhilong Qi for their significant contributions to designing, testing and providing feedback on the SDTM mapping process. We would also like to acknowledge the valuable contributions of our data scientist interns, Wenkai Xiang, Jessica Chen, and Hengjie Cui, for their exceptional programming and development of the R Shiny app.

CONTACT INFORMATION

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

Hao Chen, Yiwen Wang
AstraZeneca R&D China Biometrics
88 Xizang North Road, Jing'an District
Shanghai, China, 201210
Email: hao.chen15@astrazeneca.com
Email: yiwen.wang1@astrazeneca.com

Jaimin Patel
AstraZeneca R&D US Biometrics
Wilmington, DE 19803 United States of America
Email: jaimin.patel2@astrazeneca.com