PharmaSUG China 2026 - Paper AP-139

A Dual-Engine Architecture for MedDRA Auto-Encoding: Integrating Rule-Based Logic with LLM Semantic Retrieval in R Shiny

Qingqing Wang, Yaxiong Suo*, Xiamen Amoytop Biotech Co., LTD

ABSTRACT

Adverse event (AE) coding using the MedDRA dictionary is essential for clinical trial safety analysis but remains labor-intensive due to the dictionary's extensive scope (>60,000 preferred terms). Manual coding suffers from inefficiency and inconsistency, while existing automated approaches often lack generalizability or interpretability.

To address this, we developed a dual-engine auto-coding system through the R Shiny framework, integrating deterministic rule-based matching with large language model (LLM)-powered semantic retrieval. The rule engine prioritizes project-specific historical mappings and sponsor-defined conventions, delivering high-precision exact and pattern matching. For unmatched terms, a LLM scans the MedDRA preferred terms, enabling semantic similarity matching that returns ranked candidate codes with quantitative scores. The Shiny application supports end-to-end workflows including data ingestion, coding processing, interactive visualization, and comprehensive version comparison. Internal validation demonstrated that semantic retrieval achieved 90% accuracy, substantially reducing the need of manual dictionary lookups.

This dual-engine architecture effectively balances precision and efficiency, leveraging deterministic rule-based consistency alongside LLM semantic generalization. The framework provides a reproducible, transparent, and extensible solution for AE coding standardization, with applicability to broader clinical data transformation tasks.

INTRODUCTION

Safety evaluation is one of the core objectives of clinical trials. During trial implementation, investigators systematically collect adverse event (AE) data reported by subjects, which are typically captured as free-text exhibiting high heterogeneity and unstructured characteristics. To achieve standardized safety data exchange as required by the ICH guideline, raw adverse event terms must be mapped to a standardized medical terminology—MedDRA (Medical Dictionary for Regulatory Activities).

MedDRA, as the internationally recognized medical dictionary for regulatory activities maintained by ICH, comprises a five-level hierarchical structure: System Organ Class (SOC), High Level Group Term (HLGT), High Level Term (HLT), Preferred Term (PT), and Lowest Level Term (LLT). Accurate MedDRA coding is not only a fundamental requirement for regulatory agencies, such as FDA, EMA, and NMPA, during their review, but also serves as the data foundation for subsequent safety signal detection, benefit-risk assessment, and package insert revision.

Currently, MedDRA coding in the clinical trial industry primarily relies on manual operations where Data Managers (DMs) or medical coders consult the MedDRA dictionary term by term, completing the coding through keyword retrieval and manual interpretation. This approach presents the following significant limitations:

Efficiency Bottleneck: A typical Phase III clinical trial often involves thousands of adverse event records. Manual term-by-term coding is time-consuming and labor-intensive, with highly tedious work content that easily leads to coder fatigue and distraction.

Professional Barrier: Data managers typically lack clinical medical backgrounds. When confronting complex medical terminology, synonym variations, and spelling differences, they struggle to accurately comprehend the clinical implications of original records, resulting in coding choices that deviate from medical reality, which indicates substantial subjective variation.

Quality Control Dilemma: Traditional coding is mostly completed in spreadsheet software such as Excel, lacking systematic version control mechanisms. When data updates occur (e.g., new records added), it

becomes difficult to trace change history. Consistency verification between versions relies on manual comparison, which is both time-consuming and prone to omissions.

To address the above limitations, this system explores the integration of large language models (LLMs) into the MedDRA coding workflow and provides a function to strengthen quality control throughout the coding process.

KEY FEATURES

This system adopts R Shiny as the development framework, based on the following considerations:

Unified Deployment: Shiny applications can be deployed on internal servers, enabling centralized access and permission management across departments.

User-Friendly Interaction: Providing an intuitive web interface that lowers the barrier to using the systerm for non-technical users in daily work.

The system's core functional modules include: spreadsheet import, exact matching engine, fuzzy matching engine, version change comparison.

RULE-BASED ENGINE - EXACT MATCHING

The exact matching module enables rapid mapping for terms that are completely identical to MedDRA dictionary LLT names or conform to established coding rules. The system supports two knowledge sources:

Global Dictionary Matching: For newly initiated projects, the system directly searches the version of the MedDRA dictionary (e.g., MedDRA 27.0) uploaded for establishing precise correspondences between raw terms and LLTs.

Local Knowledge Matching: For projects with specific requirements, the system prioritizes searching the company's internal Legacy Coding Repository. This knowledge base has the standard coding results previously reviewed and confirmed by medical teams, ensuring consistency in coding practices across different studies from the same sponsor and reducing variability in coding selections for identical medical concepts.

LLM ENGINE - FUZZY MATCHING

For terms that do not achieve an exact match (such as spelling variants, synonyms, abbreviations, or non-standard medical expressions), the system prioritizes the use of the Jaro-Winkler algorithm (van der Loo, 2014; Winkler, 1990) to perform semantic similarity calculations on terms, selecting the top 50 candidate terms for subsequent LLM matching. Then the system activates a fuzzy matching engine based on LLM. This engine leverages the LLM's semantic comprehension capability to analyze the clinical significance of raw terms and retrieve semantically similar candidate terms within the MedDRA LLT hierarchy instead of regular expression which is not good at understanding clinical insight. We utilize DeepSeek V4 to achieve it. We have applied for a commerical API, which is integrated into our R Shiny code to implement LLM functionality. Regarding data security, we have implemented code-level controls to ensure that only the plain terms themselves are uploaded to the LLM server, while any sensitive information related to clinical trials is excluded, eliminating the risk of data leakage.

VERSION CHANGE COMPARISON

As clinical trial data is updated, the coding files will result in multiple versions. To quickly compare differences between successive versions, we have designed a version verification feature based on MD5 codes. When records are added, deleted, or updated, the system will flag the records and inform users.

MAIN WORKFLOW

In this section, we will introduce workflows adapted to three working scenarios to illustrate the implementation of the key features described above. The three scenarios are: coding for a new project, coding for an ongoing project, and coding for file comparison.

CODING FOR A NEW PROJECT

First, in the "Number of Candidate Matches" text box, enter how many candidate results need to be provided for each term during fuzzy matching. Then, under the "Inheritance Strategy" settings, select the "Start Over" strategy. Next, choose the exact match data source based on the actual situation. If there is a local historical coding library, it should be uploaded in advance; and then check "Custom Knowledge Base." After clicking the "Start Coding" button, the system will first perform exact match coding from the existing coding library. For records that remain unmatched, results will be generated by the LLM-based engine. Please refer to DISPLAY below.

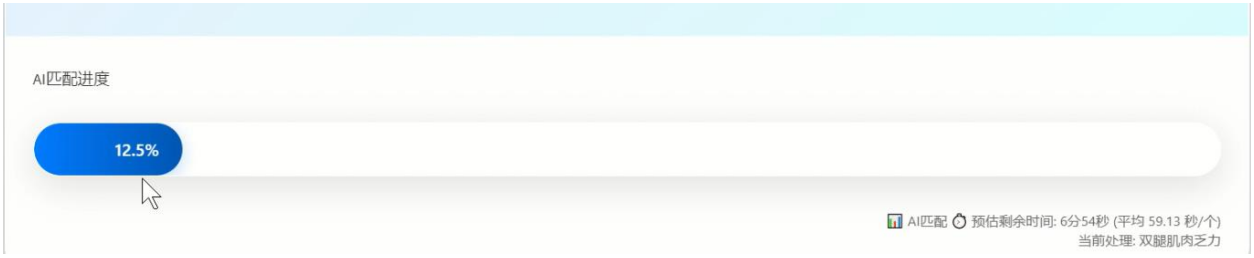
SUBJECTID	AE_NUMBER	AETERM	AESTDAT_FORMAT	AEACN_LABEL	AECON_LABEL
01001	101	发热	2024-09-11	继续用药	增加伴随用药
01001	102	头晕	2024-09-13	继续用药	未采取措施
01001	103	头痛	2024-09-13	继续用药	未采取措施
01001	104	皮疹	2024-09-17	继续用药	增加伴随用药
01001	105	白细胞计数降低	2024-09-18	继续用药	增加伴随用药
01001	106	双小腿肌肉疼痛	2024-09-26	继续用药	增加伴随用药
01001	107	乏力	2024-09-26	继续用药	未采取措施
01001	108	恶心	2024-10-01	继续用药	未采取措施
01001	109	呕吐	2024-10-01	继续用药	未采取措施

Display 1. Records exported from the EDC directly.

Display 2. Set the project name and the number of candidate terms provided by AI.

Display 3. Upload project files, standard MedDRA dictionary, and local historical coding library.

Display 4. Set the inheritance strategy and exact match strategy. Exact matching is running first.



Display 5. Fuzzy matching by LLM is running.

LLT名称	PT代码	PT名称	HLT代码	HLT名称	HLGT代码	HLGT名称	SOC代码	SOC名称	置信度	匹配说明
发热	10037660	发热	10016286	发热疾病	10005908	体温状况类	10018065	全身性疾病	1	知识库精确匹配到LLT: 发热
头晕	10013573	头晕	10029306	神经学症状	10029305	神经类疾病	10029205	各类神经系	1	知识库精确匹配到LLT: 头晕
头痛	10019211	头痛	10019233	各种头痛（	10019231	各类头痛	10029205	各类神经系	1	知识库精确匹配到LLT: 头痛
皮疹	10037844	皮疹	10052566	出疹、发疹	10014982	表皮及真皮	10040785	皮肤及皮下	1	知识库精确匹配到LLT: 皮疹
白细胞计数	10047942	白细胞计数	10047938	各种白细胞	10018851	血液类检查	10022891	各类检查	1	知识库精确匹配到LLT: 白细胞计数降低
小腿疼痛	10033425	肢体疼痛	10068757	肌肉骨骼和	10028393	肌肉骨骼和	10028395	各种肌肉骨	0.92	AI匹配: 小腿疼痛 (相似度: 0.92)
肌肉疼	10028411	肌痛	10028323	肌肉疼痛	10028302	肌肉类疾病	10028395	各种肌肉骨	0.85	AI匹配: 肌肉疼 (相似度: 0.85)
局限性肌肉	10028411	肌痛	10028323	肌肉疼痛	10028302	肌肉类疾病	10028395	各种肌肉骨	0.78	AI匹配: 局限性肌肉痛 (相似度: 0.78)

Display 6. Matching result by the dual-engine auto-coding system. The last column displays the matching method for the current record.

CODING FOR AN ONGOING PROJECT

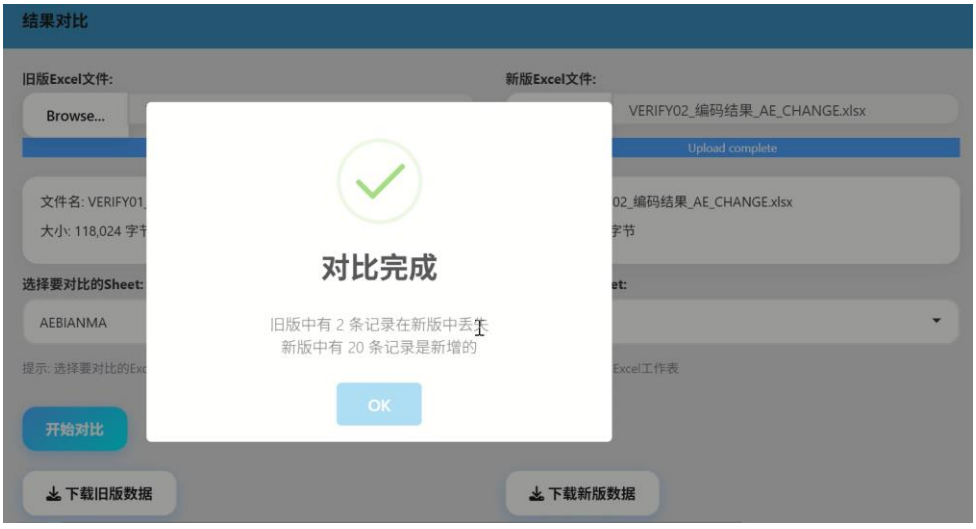
To improve coding efficiency for ongoing projects: if you already have confirmed coding results from the Nth iteration (where N ≥ 1), and need to perform the (N+1)th coding due to AE data updates, you can upload the Nth coding results to the system. The system will then execute two key logics: first, if a term in the records of the (N+1)th coding file appears in the Nth results, the coding logic for that term from the Nth iteration will be applied; second, the remaining records will be proceed by exact matching and LLM matching. Please refer to DISPLAY below.

Display 7. Upload the Nth coding result.

Display 8. The inheritance strategy is set to retrieve coding logic from the Nth result.

CODING FILE COMPARISON

You can upload the results from the Nth and (N+1)th iterations respectively, then click the "Start Comparison" button. The system will automatically check for differences between the two versions, and the following inconsistencies will be marked: newly added records in N+1; records present in N but absent from N+1; records with changed field order; records containing invisible characters. Please refer to DISPLAY below.



Display 9. File comparison page.

PT代码	PT名称	HLT代码	HLT名称	HLGT代码	HLGT名称	SOC代码	SOC名称	匹配类型	comment
10037660	发热	10016286	发热疾病	10005908	体温状况类	10018065	全身性疾病	精确匹配(知	
10013573	头晕	10029306	神经学症状	10029305	神经类疾病	10029205	各类神经系	精确匹配(知	
10019211	头痛	10019233	各种头痛（	10019231	各类头痛	10029205	各类神经系	精确匹配(知	
10037844	皮疹	10052566	出疹、发疹	10014982	表皮及真皮	10040785	皮肤及皮下	精确匹配(知	
10047942	白细胞计数	10047938	各种白细胞	10018851	血液类检查	10022891	各类检查	精确匹配(知	
10033425	肢体疼痛	10068757	肌肉骨骼和	10028393	肌肉骨骼和	10028395	各种肌肉骨	AI匹配(最佳	lost

Display 10. The last column shows the flag for suspicious records.

VALIDATION

We used a completed project to re-perform coding with this system and compared the results with manual coding. If the manual coding result fell into the candidate coding results provided by the system (2 candidates), it was considered a success; otherwise, it was considered a failure. In terms of results, the number of successes was 609 with that of failures was 61, the hit rate was 90.8%.

CONCLUSION

The dual-engine auto-coding system effectively leverages AI's powerful semantic understanding capabilities while incorporating the sponsor's existing historical coding logic, helping to improve the speed, consistency, and accuracy of clinical trial MedDRA coding. For a new project (assuming 1,000 records to be coded), it can complete the initial coding work within 1.5 hours (manual coding typically takes at least 4 hours), with an accuracy of at least 90%. However, it should be noted that the final quality control still requires human review, and this system is intended solely as an auxiliary tool. In the future, the system has significant potential for improvement in interface design and AI coding capabilities, and we remain committed to advancing these efforts.

REFERENCES

Mark P.J. van der Loo. 2014. “The stringdist package for approximate string matching” The R Journal, Vol. 6/1, June 2014, pp. 111-122.

William E. Winkler. 1990. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In Proceedings of the Section on Survey Research, pages 354–359.

ACKNOWLEDGMENTS

We would like to express our gratitude to the Rongjie Huang, coming from company's IT department, for providing server support, to Zhengfang Du and Yangyang Liu from the Data Science Department for their professional guidance about MedDRA, and to Dr.Yalin Yin from the Clinical Research Center for providing medical advice about AI prompt.

CONTACT INFORMATION

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

Qingqing Wang
Xiamen Amoytop Biotech Co., LTD
1759858051@qq.com

Yaxiong Suo
Xiamen Amoytop Biotech Co., LTD
suo11111@163.com