Paper ET06

Optimizing Clinical Research: Using AI for Automated Validation of Output Tables against ADaM

Steve Ross, Beaconcure, Raleigh, North Carolina, USA
Ilan Carmeli, Beaconcure, Boston, MA, USA

ABSTRACT

ADaM plays a crucial role in the clinical research process, by providing a structured and consistent framework for organizing and analyzing data, making it easier to generate accurate findings.

Validation of the output tables against ADaM is essential to ensure data accuracy and consistency in clinical research. By validating output tables against ADaM, clinical programmers can confidently confirm that the analysis results reflect the core data, thereby enhancing the reliability of findings in both clinical trials and research studies.

This validation process is performed primarily manually, using double programming or visual review.

In this paper, we show how this manual process can be replaced by automation using AI. We present data elements that can be derived from the output and can be matched to the ADaM standards, using multiple examples of automated validation of generated output against ADaM datasets.

INTRODUCTION

The business of clinical trials has been dramatically reshaped in our lifetime. From beginning to end, each piece of the research process – drug discovery, study design, patient recruitment, and study conduct have all been streamlined, refined, and made more efficient. So too has the statistical analysis and reporting function. Advances in data collection, statistical analysis methods, software, and computing power have reshaped what was possible even twenty years ago. Gone are the days of massive paper submissions being deposited at the offices of FDA, of non-standard datasets of myriad formats, and waits of a year to hear whether your NDA has been successful. However, one thing has not changed over time – the historically resource-intensive process of TLF validation.

From the point where TLF programming begins, multiple organizational functions work together to program, validate, and review the display set to ensure that the final deliverable is fit for purpose, and analyses demonstrate the 'truth' of the data contained in the ADaMs. Programmers perform double programming to generate and then validate the body of each table with PROC COMPARE, and the same programmers also use visual validation to perform still other checks. Similarly, additional reviewers from sponsors, statistics, clinical, PK, medical writing, pharmacovigilance, and others also spend hours poring over tables, listings and figures, lending their expertise over and over again in the course of a study to deliver the most high quality product to an internal client, a sponsor, or a regulatory body.

Today, the promise of AI is enabling the next generation of advances in clinical trial analysis, reporting and validation. Machine learning and natural language processing allow us to ingest large volumes of data, creating a database of connections in structure and content of outputs and ADaM datasets, and automating validation tasks to run more quickly, efficiently, and accurately than ever before. This is accomplished with a multifaceted approach, using metadata and data mined from the TLFs themselves, as well as tables of contents and mock shells, and using these to link directly back to the ADaM datasets. ADaM datasets themselves are programmed with the end in mind ('One PROC away') and designed to have a standard, repeatable structure that lends itself well to automated analysis of relational databases commonly found in clinical trials data. Paired with specification documents and even other TLF outputs, it is possible to link and validate these displays using AI to 'connect the dots.'

SOMETHING OLD, SOMETHING NEW IN CLINICAL TRIAL VALIDATION

Despite industry-wide efforts to shorten, streamline, standardize, and reduce the outputs in the typical clinical trial, advancements in technology (even aside from the sheer 'throughput' increases in processing power) have created an environment where pharmaceutical research casts a wider and deeper net in search of data. The application of concepts such as adaptive design and simulation (statistics) and the introduction of big data/real world evidence/real world data (data science) into the pharmaceutical research vocabulary have proliferated of late in an effort to hasten the discovery, development and marketing of life-saving new treatments much more quickly than even a decade ago.

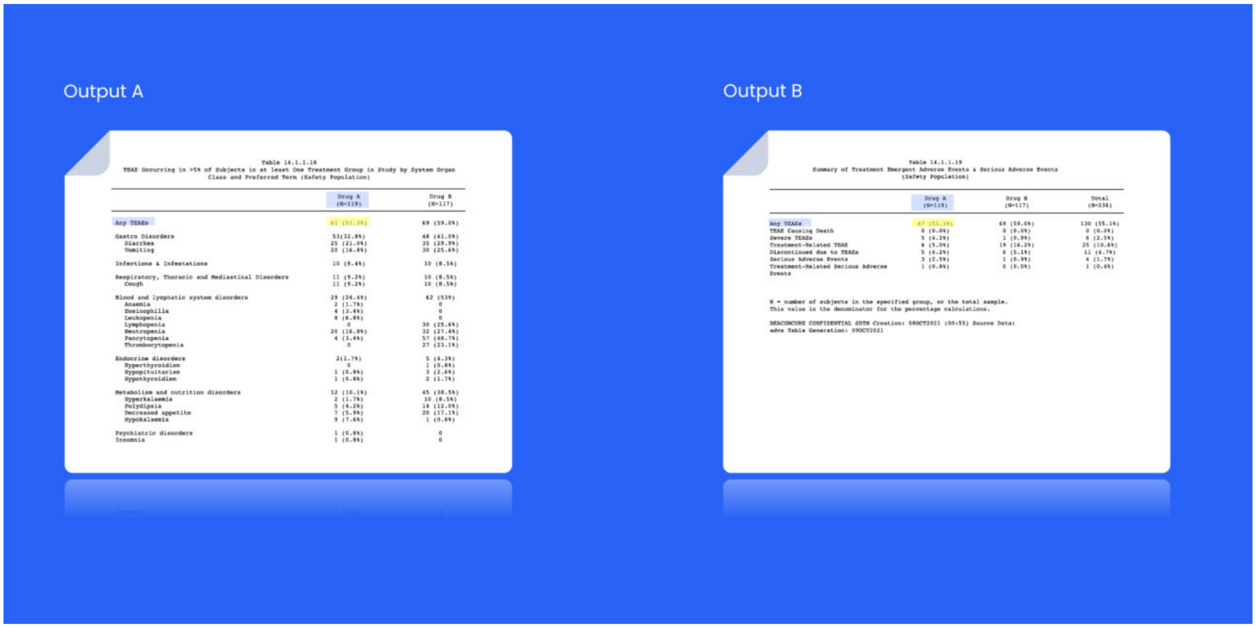
The inherently inefficient “duplication by design” nature of the current validation “gold standard” of double programming can be dramatically simplified, improved, and made faster using current machine learning (ML) and natural language processing (NLP) methodology. Capitalizing on the best features of human and machine skill sets (understanding and speed, respectively), we can realize the twin goals of making the analysis and reporting function faster and, at the same time, more robust. When using Verify during TLF programming and validation, AI processing can perform repetitive *and intuitive* checks on large TLF sets. This can be accomplished orders of magnitude faster and more accurately than humans, checking and rechecking aspects such as titles and footnotes, hierarchical checks to check for decreasing n’s, counting, and even cross-checks across displays, as a deliverable comes closer to completion. ML and NLP enables users to make and use connections and linkages to automate these aspects of the current common practices, some of which are programmed, and some are performed manually.

This does not imply that human intuition, understanding and interpretation are being replaced. To the contrary, here the human element is paired more elegantly with the machine, in a system that plays to the strengths of both. The statistician-designed specifications and programmer-created displays are used by the AI machine to create a metadata store used in data acquisition to subset and compare to the ADaMs. In short, a human generates a table using a specification, and the machine (Verify ML, NLP processes) uses the metadata captured from the TLFs and mock shell specifications, to link to variables in ADaMs. In doing so, the TLF is still the ‘source’ or ‘referent’, and is the ‘PROD’ side of the PROC COMPARE. The ‘VAL’ side of the COMPARE is arrived at with a combination of AI-extracted metadata and the ADaM datasets themselves. Verify software performs checks that are both programmatic AND visual. Similarly, checks on the output TLF(s) use the table set itself AND the ADaM data.

The advantages are most clearly apparent as a deliverable enters its critical stage (after a DBL event). Programmers are updating TLFs based on feedback from reviewers (and visually and programmatically validating them!), and statisticians are checking to make sure that the updated values make sense in context. Multiple iterations are often necessary due to data, formatting, renumbering, or other unforeseen issues. As the deliverable deadline looms, visually validating and revalidating large tables (or sets of tables) becomes cumbersome and onerous, consuming human resources that might be better deployed in other areas of the TLF package. The use of AI in this context makes great sense. Let statistics and programming focus on the understanding and intuitive aspects of the displays, while using the machine to churn through checking that 300 table titles and footnotes still match a specification, that all tables in a deliverable are still present and accounted for, and that non-programmed but intuited visual validation checks are in place to ensure a quality deliverable. An added bonus is that once the TLF and ADaM data are reflected in the AI model, they become part of the ‘knowledge base’ of study information, and can be used and reused to compare to a single output, multiple outputs, and even across multiple study outputs. This is particularly impactful in the context of multiple study integrations, from DSURs to NDA submissions.

MAKING CONNECTIONS

In Figure 1 below we illustrate this concept.



Output A

Table 14.1.1.16
TEAE Occurring in >5% of Subjects in at least One Treatment Group in Study by System Organ Class and Preferred Term (Safety Population)

	Drug A (N=119)	Drug B (N=117)
Any TEAEs	61 (51.3%)	69 (59.0%)
Gastro Disorders	53(32.8%)	48 (41.0%)
Diarrhea	25 (21.0%)	35 (29.9%)
Vomiting	20 (16.8%)	30 (25.6%)
Infections & Infestations	10 (8.4%)	10 (8.5%)
Respiratory, Thoracic and Mediastinal Disorders	11 (9.2%)	10 (8.5%)
Cough	11 (9.2%)	10 (8.5%)
Blood and lymphatic system disorders	29 (24.4%)	62 (53%)
Anaemia	2 (1.7%)	0
Eosinophilia	4 (3.4%)	0
Leukopenia	8 (6.8%)	0
Lymphopenia	0	30 (25.6%)
Neutropenia	20 (16.8%)	32 (27.4%)
Pancytopenia	4 (3.4%)	57 (48.7%)
Thrombocytopenia	0	27 (23.1%)
Endocrine disorders	2(1.7%)	5 (4.3%)
Hyperthyroidism	0	1 (0.8%)
Hypopituitarism	1 (0.8%)	3 (2.6%)
Hypothyroidism	1 (0.8%)	2 (1.7%)
Metabolism and nutrition disorders	12 (10.1%)	45 (38.5%)
Hyperkalaemia	2 (1.7%)	10 (8.5%)
Polydipsia	5 (4.2%)	14 (12.0%)
Decreased appetite	7 (5.9%)	20 (17.1%)
Hypokalaemia	9 (7.6%)	1 (0.8%)
Psychiatric disorders	1 (0.8%)	0
Insomnia	1 (0.8%)	0

Output B

Table 14.1.1.19
Summary of Treatment Emergent Adverse Events & Serious Adverse Events
(Safety Population)

	Drug A (N=119)	Drug B (N=117)	Total (N=236)
Any TEAEs	67 (51.3%)	69 (59.0%)	130 (55.1%)
TEAE Causing Death	0 (0.0%)	0 (0.0%)	0 (0.0%)
Severe TEAEs	5 (4.2%)	1 (0.9%)	6 (2.5%)
Treatment-Related TEAE	6 (5.0%)	19 (16.2%)	25 (10.6%)
Discontinued due to TEAEs	5 (4.2%)	6 (5.1%)	11 (4.7%)
Serious Adverse Events	3 (2.5%)	1 (0.9%)	4 (1.7%)
Treatment-Related Serious Adverse Events	1 (0.8%)	0 (0.0%)	1 (0.4%)

N = number of subjects in the specified group, or the total sample.
This value is the denominator for the percentage calculations.

BEACONCURE CONFIDENTIAL SDTM Creation: 08OCT2021 (00:55) Source Data:
adve Table Generation: 09OCT2021

Figure 1: Two sample AE tables with discrepancy highlighted

Here we present two of a common sequence of AE tables, generated by our PROD programmer. In the current validation model, a second (validation) programmer, using the same TOC, table specification, and ADaM data programs the same analysis, and performs both visual review and programmatic validation, as a check on the first programmer.

Visual review might entail checking that the titles and footnotes match the specification and are not truncated or wrapped, check that n's are generally decreasing over time in a visit-based display, or spot checking that horizontal/vertical counts sum appropriately. These types of checks are either cumbersome and time-consuming to program or require a deeper understanding of the outputs, and instead are visual checks done by a statistician or programmer. Depending on the display, there might be a dozen other unprogrammed additional checks that visual validation would pick up on but programmatic validation would not. Some might take just a minute or two, but aggregated over multiple iterations, deliverables, and a small army of reviewers (clinicians, medical writers, kineticists, et al), the time adds up, especially the closer a deliverable date approaches.

Programmatic validation uses ADaM data to independently generate counts, percents, univariate statistics, p-values, et cetera, for the body of a given table. At most basic, this is the function double programming performs and at similarly high time investment.

In the above example, cross-table validation is performed visually, with the programmer and/or statistician checking that the values in the 'unique' table (the first in a series of related tables) match those appearing in the 'repeat' table. In the Verify model, the AI-created metadata links created during the 'training' of the model allow us to do this programmatically, and irrespective of the size of the table or the number of tables in the series.

It is important to note here the distinction between using a set of digitized TLF data to validate tables, and using ADaM data. In a traditional double programming environment, both production and validation programmers use ADaM data. In contrast, using AI, the source data are the digitized TLF data augmented by ADaM data. Verify deploys a hybrid method of validation, which has the ability to use both the TLF data and the underlying ADaMs to validate the accuracy of the outputs.

Figure 2:

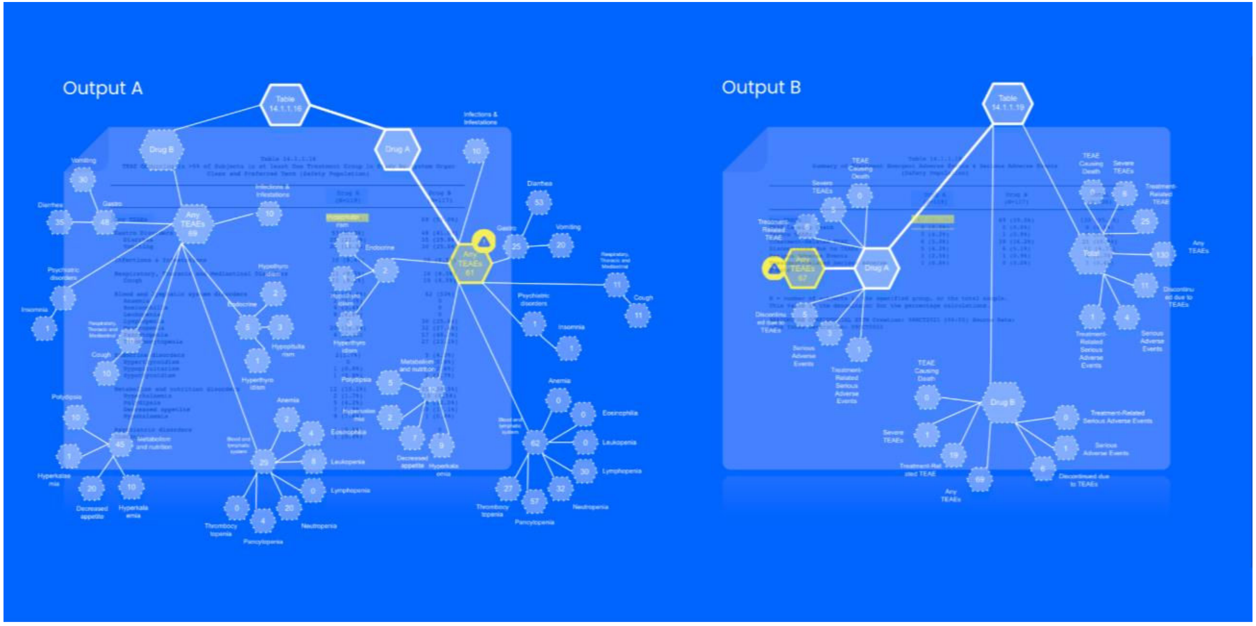


Figure 2: Representation of TLF metadata extracted for analysis and validation

DEEP DIVE

Each of the tables in Figures 1 and 2 contain key pieces of information (i.e., metadata). Verify algorithms ‘harvest’ these key pieces of information into a robust structured database for the purpose of validation. For example, the titles are broken down into an AI representation of entities such as population and treatment assignment. This allows the system to query the output tables and cells that relate to the same clinical issues. Once these are identified, the Verify algorithm determines the corresponding columns and rows in the ADaM datasets. The system then performs the required statistics (such as N, n, mean, SD, %) on the data from the ADaM datasets in order to validate the data in the TLFs against the Verify-programmed numbers.

As a result, the validation is generalized, and not limited to a specific set of initial tables or specs, and can instead be applied to understand any table introduced at any stage of development. In addition, an AI model can be easily adapted to new data, such as new formats and new logical groupings. By comparison, tools that are developed without ML/AI to do similar things (such as R and SAS), can be inflexible and limited due to strict information and output placing requirements.

Here we have discussed single table vs. ADaM examples, but the larger application possibilities become apparent. We have now created a robust AI-enabled database that can be used in multiple ways – serving to perform checks as both a visual validator and a programmatic one.

CONCLUSION

The validation process in clinical research is comprehensive, thorough, and exacting. It is also by design duplicative, repetitive, and time-consuming. It does not have to be. Advancements in AI technology offer a hybrid validation methodology that uses both human and machine strengths to arrive at an intuitive, repeatable, and robust process for validation of TLFs that rivals the efficacy of double programming.

ACKNOWLEDGMENTS

The authors would like to thank Achinoam Ravet Perel, Sarah Michaels, and Sharon Atzmon for their invaluable assistance in the drafting and review of this paper.

RECOMMENDED READING

https://beaconcure.com/

CONTACT INFORMATION

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

Author Name: Steve Ross
Company: Beaconcure
Address: 117 Kendrick St, Suite 300, Needham, MA 02494
Email: steve@beaconcure.com
Website: www.beaconcure.com
LinkedIn: http://www.linkedin.com/company/beaconcure/