Paper ET-14

PHUSE US Connect 2026

AI Reading and Understanding the SAP to Generate the TFL TOC

Toshio Kimura, Arcsine Analytics
Siqi Wang, Arcsine Analytics
Weiming Du, Alnylam Pharmaceuticals
Songgu Xie, Regeneron Pharmaceuticals

ABSTRACT

What if AI can read and understand the SAP and learn about endpoints, analysis sets and analysis methods? Can AI then generate the TFL TOC?

We developed a TOC generating AI solution using Python and ChatGPT. After uploading the SAP, AI agents collect all SAP components and generates the TOC. The solution consists of:

1) Document processing: Prepare AI-ready document
2) Metadata collection: AI agents collect SAP metadata
3) TOC generation: Using collected metadata, AI generates TOC

We will demonstrate an AI solution generating the TOC. Key learnings include:

1) Large context: Processing large context documents such as the SAP requires a different more scalable AI strategy.
2) Specialized AI agents: We developed specialized AI agents specifically designed to collect different SAP metadata.
3) Hybrid AI and traditional programming: AI combined with traditional programming yielded the best results.

If AI can generate the TOC … what else can it do?

INTRODUCTION

While it has been a few years since the release of generative AI models such as ChatGPT, AI has not yet truly revolutionized the field of biostatistics and statistical programming. Within the field of biostatistics and statistical programming, AI thus far has been used mainly as a coding assistant. While the performance boost from AI code assistants is not insignificant, it is hardly a revolution or a reimaging of the entire field.

The key to a bigger leap forward starts with a bolder, more ambitious vision. The true potential of AI has been hampered not just by technology but rather by the limited scope of the use cases pursued. The scope has been limited to individual TFLs and analysis. But, what if we put forth a much more ambitious goal of understanding and executing the entire SAP? What if we can upload the SAP and receive back all of the TFLs as the output? That would be a true revolution. We developed an agentic AI proof-of-concept (PoC) solution that will eventually bring about this revolution.

SCOPE

While the ultimate goal of AI understanding the SAP is to automatically generate the TFLs, the scope of this PoC was limited to generating the TFL table of contents (TOC) as the first step. The PoC was also further limited to developing the TOC based on the statistical analysis that are explicitly specified as there are times when TFLs are included in the TOC even though they are technically not specified in the SAP.

OVERVIEW

We used an agentic AI framework to develop this solution. In order to develop an agentic AI solution to understand the SAP and automatically generate the TFL TOC, 3 major steps are required:

1) Document ingestion and processing: SAP contents were broken up into each section. This allows the document to be AI-ready where various SAP sections can be easily queried and used as an input into the LLM.

2) Metadata collection: The main analysis components required for the TFL TOC are extracted using specifically designed AI agents. These include analysis sets, endpoints and statistical analysis methods and the relationships between these components.

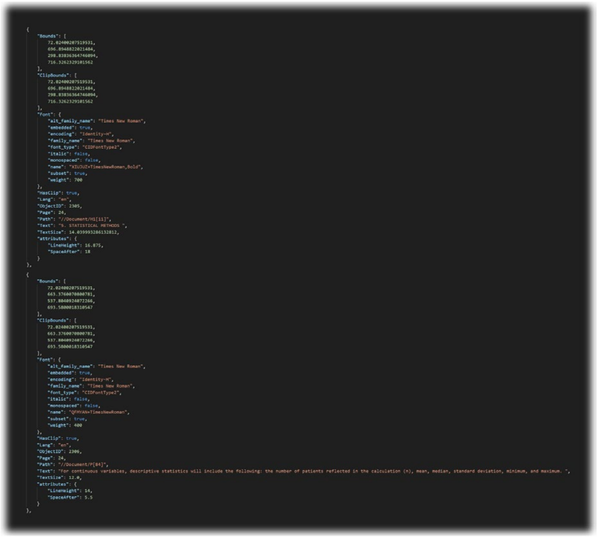
3) TOC generation: Using the collected metadata, the AI solution generates the TFL TOC.

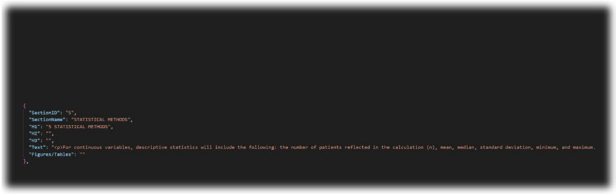
DOCUMENT INGESTION AND PRE-PROCESSING

The very first step of this process is to ingest and pre-process the SAP into an AI-ready format. Once the SAP is uploaded, the document will be processed to break up the content from each section and stored in a database. Keeping each section as close to the original source as possible while adding structure will allow for greater flexibility for downstream use.

While several different methods were explored, the Adobe Extract API provided the highest quality in preserving the original content and structure. However, the API returns every detail of the document. Therefore, the returned object must be processed to keep only the relevant information.

Figure 1 below shows the raw output of the Adobe Extract API (Figure 1A) along with the clean output that is stored in the database (Figure 1B).





A: Raw output from Adobe Extract API B: Clean AI-ready output

Figure 1: Pre-Processing SAP to be AI-ready

With the SAP content in a database, various sections and analysis components can now be easily queried and used for downstream processing. Table 1 below shows how the SAP sections are stored in the database.

Table 1: SAP content stored in the database

SectionId	SectionName	Content
5	Statistical Methods	For continuous variables, descriptive statistics will include the following: the number of patients reflected in the calculation (n), mean, median, standard deviation, minimum, and maximum. For categorical or ordinal data, frequencies and percentages will be displayed for each category.
5.1	Demographics and Baseline	Demographic data and baseline characteristics variables will be summarized, by treatment group and overall, using descriptive statistics.
…	…	…

METADATA COLLECTION

Once the SAP is broken up and stored in a database, the relevant sections can be queried and submitted to the LLM in order to extract key analysis metadata information. The extracted information is returned using a defined structured data schema. Since the information content extracted for each component differ widely, separate AI agents were developed specifically tailored to each component while the platform orchestrates the separate AI agents.

The following steps outline the workflow used to extract key content:

1) Query SAP content: The LLM processes the SAP TOC to identify the relevant sections related to each component (e.g., endpoints, population set, statistical analysis method, etc.). This information is used to query the database to pull the relevant SAP section content that will be used by various AI agents.

2) Extract key analysis metadata information: For each component, separate AI agents were developed to extract key analysis metadata information. For example, one AI agent would evaluate the analysis set section and extract the analysis set information into a structured data schema. Similarly, another AI agent would evaluate the statistical analysis method section and extract the statistical analysis method information into a structured data schema. This is done for all of the components required for the TFL TOC.

For each component, separate AI agents were developed. However, these AI agents follow a common structure:

1) General: There are general overarching rules that are submitted as part of the prompt for all AI agents.

2) Component specific:

    a. Component specific examples: The few shot prompting method was used. Therefore, multiple sets of example input and output pairs were prepared for each component from other SAPs.

    b. Component specific output JSON data schema: Pre-defined output JSON data schema and associated rules were developed as the return object of the LLM. The few shot examples also follows this output data schema.

    c. Component specific rules: Component specific rules were developed to provide the LLM with additional rules to follow.

Other tweaks were made to optimize the quality of the data extraction. Some challenging situations required breaking up the evaluation into multiple steps and multiple passes through the LLM. Examples of challenging situations include the following:

1) Ambiguous timepoints

    Secondary endpoint: >= 10 point improvement at weeks 24 and 48

    Interpretation 1: >= 10 point improvement at <BOTH> weeks 24 and 48
    Interpretation 2: >= 10 point improvement at weeks 24 and 48 <SEPARATELY>

    Most would agree that Interpretation 2 is the more likely intention. However, Interpretation 1 is also technically correct (for example, if the endpoint in question was referring to sustained response requiring response over multiple timepoints).

2) Ambiguous endpoints (multiple endpoints)

    Secondary endpoint: Fluid-free retina (total fluid, IRF, SRF at foveal center and in the center subfield)

Interpretation 1: Fluid-free retina is a new composite endpoint where there are no fluids in each of the defined areas as indicated in the parenthesis.

Interpretation 2: Fluid-free separately for each type within the parenthesis:
- Fluid-free total fluid
- Fluid-free IRF
- Fluid-free SRF at foveal center
- Fluid-free SRF in the center subfield

Within the ophthalmology therapeutic area, Interpretation 2 is the correct interpretation.

3) Ambiguous endpoints (multiple criteria)

Secondary endpoint: Proportions of patients gaining and losing ≥ 5 or ≥ 10 letters

Interpretation 1: Perhaps this endpoint is to capture high variability. Therefore, in order to meet this definition a patient must both gain >=5 (or >=10) letters and lose >=5 (or >=10) letters during the study.

Interpretation 2: Multiple endpoints are defined:

1) Gaining >=5 letters
2) Gaining >=10 letters
3) Losing >=5 letters
4) Losing >=10 letters

Within the ophthalmology therapeutic area, Interpretation 2 is the correct interpretation.

In order to increase transparency and traceability and minimize hallucinations, we incorporated additional checks and flags. For example, we flag higher risk data points. When AI is confronted with non-standard wording or somewhat vague language or if we get different responses from multiple runs, we flag that data point for human review. We also document whether the data point came directly from the SAP or if it was inferred as well as capturing the verbatim text source that was used in the evaluation.

TOC GENERATION

After the analysis metadata have been extracted, the AI solution will need to generate the TOC. However, in order to develop the TOC, the relationship across the various analysis components must be established. For example, the system will have to determine the analysis sets and endpoints that are associated with various statistical analysis methods as not every combination would be appropriate. The system will also determine the relative ordering of the endpoints and methods so that they appear in the correct order within the TFL TOC.

Similar to the AI agents that extract analysis metadata as described above, a dedicated AI agent was developed to determine the relationships between components. Rather than identifying various analysis components, this AI agent focuses on identifying the relationships between components. After identifying the relationships and relative ordering, the AI agent then generates the TFL TOC.

The system will not only output the final TFL TOC, but it will also output all of the extracted analysis metadata including the flags. These can be reviewed by human reviewers and updated if necessary. A future enhancement will be a method for the human reviewed input to be automatically incorporated back into a feedback loop, so that the agentic AI system can learn and incorporate human feedback into future processing.

CONCLUSION

Our successful PoC demonstrates that AI is capable of understanding and ultimately executing the entire SAP. We deployed an agentic AI framework that orchestrates the work of multiple AI agents each tasked with a specific purpose. Another AI agent then pulls everything together to generate the final TFL TOC as the final output.

The TFL TOC that it generates also contains more than a standard TFL TOC. It contains comprehensive analysis metadata sufficient to execute the analysis. Therefore, the TFL TOC is a data-rich enhanced TOC that can be used directly by downstream systems for execution thus further supporting our vision for an agentic AI driven execution of the SAP.

CONTACT INFORMATION

Contact the author at:

Author Name:	Toshio Kimura
Company:	Arcsine Analytics
Email:	toshio.kimura@arcsineanalytics.com