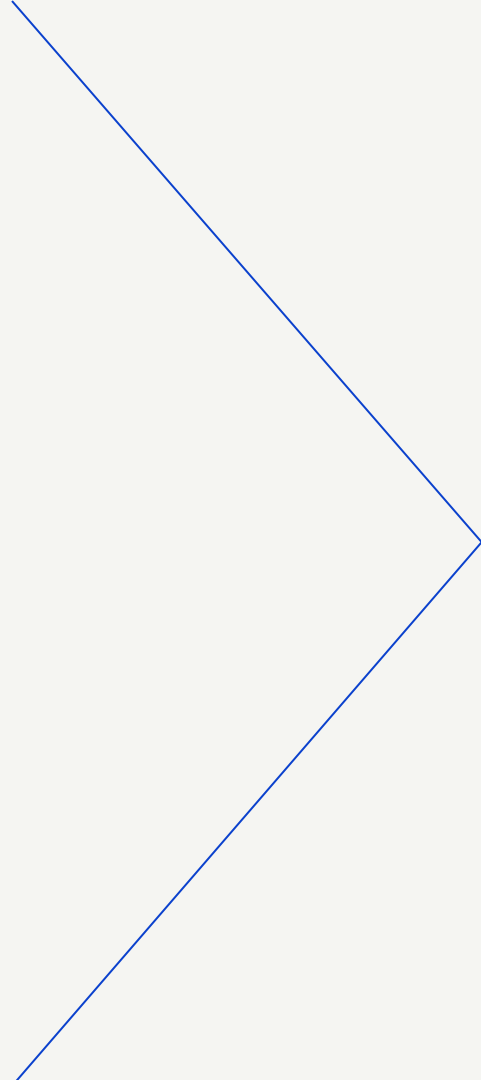


Integrating GenAI into Clinical Insights Generation to Boost Data Science Efficiency

Our Journey So Far, Reflections, & Future Outlook

Vincent Shen
Senior Principal Data Scientist
Roche

Agenda

- 
- A decorative graphic consisting of a vertical blue line and two diagonal blue lines that meet at a point on the vertical line, forming a large right-pointing arrow shape that spans the height of the slide.
- 1. Business Context & Problem Statement**
 - 2. GenAI Products & Business Impact**
 - 3. Reflections on Our Journey So Far**
 - 4. Vision on GenAI for Insights Generation**

Business Context & Problem Statement

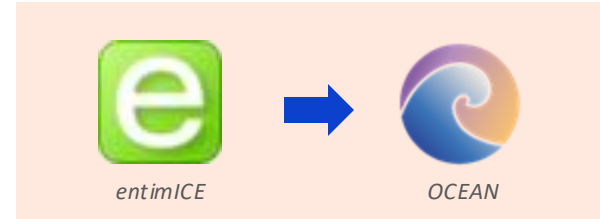
Perspectives from 1 year ago

Business Context - From 1 Year Ago

Why did we do this?

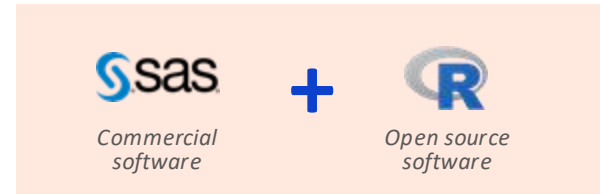
New platform

Migrating all clinical studies from entimICE to OCEAN



New programming languages

From SAS to SAS+R+more



New processes & tools

Git workflow, GitLab, R packages for SDTM/ADaM/TLG



- ➡ Significant efforts & resources required for change management & user support
- ➡ Steep learning curve on new tools, despite the documentation & training materials

How Did We Approach the Problem Back Then?

Why GenAI/LLM?

- ➔ Significant efforts & resources required for change management & user support
- ➔ Steep learning curve on new tools, despite the documentation & training materials

- GenAI was a fast-growing field and had generated a great level of interest globally
 - **And it still is today**

- GenAI/LLM had been shown to have the following capabilities:
 - Good performance on extracting information from domain-specific contexts and generating answers
 - Relatively low cost on time & effort to build a domain-specific app
 - Communicate like a human

- These capabilities made GenAI/LLM a natural choice to solve our business problems
 - **And we still feel the same today**

GenAI Products & Business Impact

Development of OCEAN Assistant & NExpert

OCEAN Assistant

A GenAI-based chatbot to provide real-time support on OCEAN-related questions

Design Architecture

- RAG (Retrieval Augmented Generation)-based application with GPT-4o
- 20+ sources of documentation on various systems, tools, processes
- Semi-automated data pipeline to fetch latest information from various sources
- Traceability of answers & self-evaluated confidence score on answers provided

Business Outcomes

- Serve as an **automatic** first-line user support & reduce the workload on SMEs
- **Lessen the learning curve** for OCEAN & related tools
- Make the 20+ sources of documentation **more digestible** for users
- Help users to **troubleshoot** complex processes & production runs

Metrics

1,000+

active users (including
ExBPs)

44,000+

questions asked &
answered

4,300+

production runs
debugged

~25

hours saved* per 100
questions

* Assuming on average, it takes ~15 min to answer any question (including documentation search, find & chat with SMEs, etc.)

OCEAN Assistant

Some use cases

■ Retrieve documentation

■ *What is the orc command on OCEAN?*

 What is the orc command on OCEAN?

 The `orc` command on OCEAN is an application designed to support Ocean Orchestration. It allows users to execute and manage IO pipelines, as well as set up or manage scheduled IO pipelines. The command syntax is as follows:

```
orc [OPTIONS] COMMAND [ARGS] ...
```

Options

- `--help`: Shows help message and exits
- `--version`: Information about the current Orchestration Client version
- `--install-completion`: Install completion for the current shell
- `--show-completion`: Show completion for the current shell, to copy it or customize the installation

Commands

- `configure`: Creates a configuration file that enables connecting to the orchestration system.
- `cron-helper`: Prints cron syntax explanation.
- `delete-schedule`: Deletes a scheduled pipeline.
- `edit-schedule`: Edits a scheduled pipeline.
- `inspect-pipeline`: Inspects a specific pipeline.
- `job-logs`: Retrieves logs for a specific job.
- `list-pipelines`: Lists available pipelines.
- `list-schedules`: Lists scheduled pipelines.
- `run-pipeline`: Executes a pipeline.
- `schedule-pipeline`: Schedules a pipeline for execution.

To use the `orc` tool, it must first be configured using the `orc configure` command, which involves creating a Personal Access Token in GitLab and entering it when prompted.

Confidence:  High

Documentation links: [io-ocp-tools](#) • [list-of-commands](#) • [tools-ocp-tool](#)

Ask a question. Provide as much details/code as possible

OCEAN Assistant

Some use cases

- **Generate simple snippet of code**

- *How to create AE domain with roak?*


 How to create AE domain with roak?


 To create an AE domain using `roak`, you can follow these steps based on the provided context:

- Load Study Configuration:** First, ensure that you have loaded the study configuration files into your `oak_env`. This is typically done using the `oak_load_study_config()` function.


```
path_ep <- system.file("extdata/ep_study/study_configuration_files", package = "roak")
oak_load_study_config(path_ep, oak_env = oak_env)
```
- Create the AE Domain:** Use the `create_domain()` function to create the AE domain. You need to specify the domain name, the metadata, and any other necessary parameters like `epoch_date_var` and `use_vstp`.


```
ae <- create_domain(
  "AE",
  sdtmv_spec_metadata = sdtmv_spec_metadata,
  epoch_date_var = "AESTDTC",
  use_vstp = FALSE,
  oak_env = oak_env
)
```
- Add the AE Domain to `oak_env`:** After creating the domain, add it to the `oak_env` using the `oak_set_sdtmv_domain()` function.


```
oak_set_sdtmv_domain(domain_name = "AE", oak_env = oak_env)
```
- Handle Missing Variables:** If there are any missing variables in the AE raw dataset that are required for the SDTM TS domain, you may need to add them manually. For example:


```
oak_env$raw_dataset$AESAEVERSN <- "MedDRA 26.0"
oak_env$raw_dataset$AESCMDICT <- "WHODrug GLOBAL 83 MARCH 1, 2023"
```

By following these steps, you should be able to create and configure the AE domain within your `oak_env` using `roak`.

Confidence: 🟡 Medium

Ask a question. Provide as much details/code as possible
 

OCEAN Assistant

Some use cases

■ Solve potential errors you encounter

■ WorkflowError:

MissingInputException: Missing input files for rule adsl


I have this error in snakemake: WorkflowError: MissingInputException: Missing input files for rule adsl



The error you're encountering, `MissingInputException: Missing input files for rule adsl`, indicates that Snakemake cannot find one or more of the input files specified for the `adsl` rule. Based on the provided context, the `adsl` rule requires the following inputs:

- `programs/ad_adsl.R`
- `data/dm.parquet`

To resolve this issue, you should:

- Verify File Existence:** Check if both `programs/ad_adsl.R` and `data/dm.parquet` exist in the specified directories. If any of these files are missing, Snakemake will not be able to proceed with the `adsl` rule.
- Check File Paths:** Ensure that the paths to these files are correct in your Snakemake file. Any typo or incorrect path will lead to this error.
- Dependency Rule:** If `data/dm.parquet` is supposed to be generated by another rule (e.g., the `dm` rule), make sure that rule is correctly defined and executed before the `adsl` rule. Snakemake should automatically handle this if the rules are set up correctly.
- File Generation:** If `data/dm.parquet` is an output from another process or rule, ensure that process has been completed successfully and the file is generated.
- Check Timestamps:** If the files exist but are not being recognized, check if there are any issues with file timestamps that might be causing Snakemake to think they are outdated or missing.

By following these steps, you should be able to identify and fix the issue causing the `MissingInputException` for the `adsl` rule.

Confidence: ● High

Documentation links: [io-error-guide](#) • [glopo](#)


Overflow posts

Ask a question. Provide as much details/code as possible

OCEAN Assistant

Some use cases

- **Retrieve from Roche Overflow posts**
 - *Do I need to manually set dictionary versions when creating TS domain?*

 Do I need to manually set dictionary versions when creating TS domain?

 No, you do not need to manually set dictionary versions when creating TS domains. The provided context shows that the dictionary versions are not manually set during the creation of the TS domain or any other domains. The process involves using functions like `create_domain_tdd()` and `oak_set_sdtav_domain()` to create and set the domains within the `oak_env` environment.

Links to UGO: [roak link](#) • [roak link](#) • [Onboarding Exos](#)

Overflow posts
⌵

Dynamic dictionary setting in SDTM domain TS?

roak • <https://overflow.roche.com/t/18939>

@Tim_Barnett - In roak, for TSPARMCD= `AEDICT`, latest values will be dynamically fetched from `AED$AEVERSN` variable for LP studies and `AESAEVERSN` variable for EP studies.

Similarly for TSPARMCD= `CMDICT`, latest values will be dynamically fetched from `MD$CMDICT` variable for LP studies and `AESCMDICT` or `CM$CMDICT` variable for EP studies.

Also oak will be producing a note in console to inform users about the source of dictionary values as mentioned above.

Thus in SDTM TS domain, dictionary values will always be latest and fetched from DCH source data.

OCEAN Assistant

Some use cases

- **Since it's LLM-based, you can also use it for other purposes ...**
 - *Can you create a quiz question related to tools and processes in OCEAN?*

 can you create a quiz question related to tools and processes in OCEAN? Highlight the correct answer.

 Here is a quiz question related to tools and processes in OCEAN:

What is the primary function of the OCEAN Assistant?

- A) To generate code for new projects
- B) To perform question/answer tasks using OCEAN's related documents
- C) To manage user accounts
- B) To perform question/answer tasks using OCEAN's related documents

NExpert

A GenAI-based chatbot to help programmers customize admiral/NEST code when creating ADaMs/TLGs

Design Architecture

- Chain-of-Thoughts (langchain) structure with GPT-4o + fine-tuned open-source model
- Package documentation + curated/well-documented code examples as source data
- Code review functionality to ensure the quality of the code generated by AI
- Semi-automated pipeline to retrieve information from package manual & examples as source data for the chatbot

Business Outcomes & Benefits

- **Lessen the learning curve** for {admiral}, {rtables}, {tern}, etc.
- Serve as the first-line support to help users to address their ADaM & TLG **customization needs**
- Provide human-readable **explanations** to the recommended code
- Help users to **troubleshoot** common errors when using selected R packages

NExpert

Some use cases

■ Customization with {rtables} & {tern}



Here is my table. I only want to keep rows that have a number > 10% in any of the columns.

```

result <- basic_table() %>%
  split_cols_by("TRT01P", split_fun = add_overall_level("Total", first = FALSE)) %>%
  analyze_vars("RACE", .stats = "count_fraction", na_str = NA_character_) %>%
  build_table(adsl)

result

```



To keep only the rows where the count fraction is greater than 10% in any of the columns, you can use the `prune_table` function with the `has_fraction_in_any_col` condition. Here's how you can modify your code:

```

result <- basic_table() %>%
  split_cols_by("TRT01P", split_fun = add_overall_level("Total", first = FALSE)) %>%
  analyze_vars("RACE", .stats = "count_fraction", na_str = NA_character_) %>%
  build_table(adsl)

# Prune the table to keep only rows with count fraction > 10% in any column
row_condition <- has_fraction_in_any_col(atleast = 0.10)
result <- prune_table(result, keep_rows(row_condition))

result

```

This code will filter the table to include only those rows where the count fraction exceeds 10% in any of the treatment columns.

Like this answer?

☐ YES
 ☐ NO

Ask your questions here

NExpert

Some use cases

■ Customization with {admiral}

 Add a new variable `FXENDTM` to the datepart of `FX.FXENDTC` using `FX` domain, `date_imputation = "first"`, `time_imputation = "first"`. If `FXOUT == "FATAL"`, then the highest imputation will be `"Y"`, `min_date` as `AESTDTM`; if `FXOUT != "FATAL"`, the highest imputation will be `"M"`.

 To add a new variable `FXENDTM` to the date part of `FX.FXENDTC` using the `FX` domain with the specified imputation rules, you can use the `slice_derivation` function along with `derive_vars_dtm`. Here's how you can implement this:

```
fx <- fx %>%
  slice_derivation(
    derivation = derive_vars_dtm,
    args = params(
      dtc = FXENDTC,
      new_vars_prefix = "FXEN",
      date_imputation = "first",
      time_imputation = "first"
    ),
    derivation_slice(
      filter = FXOUT == "FATAL",
      args = params(
        highest_imputation = "Y",
        min_dates = exprs(AESTDTM)
      )
    ),
    derivation_slice(
      filter = FXOUT != "FATAL",
      args = params(
        highest_imputation = "M"
      )
    )
  )
```

This code will create the `FXENDTM` variable with the specified imputation rules based on the value of `FXOUT`. If `FXOUT` is `"FATAL"`, the highest imputation level will be `"Y"` with a minimum date of `AESTDTM`. Otherwise, the highest imputation level will be `"M"`.

Reflections on Our Journey So Far

What have we observed since launching the two solutions? How does the evolution of GenAI change our views?

Reflection on Knowledge Retrieval Tasks

With ever-increasing data sources, is our approach scalable?

Key observations:

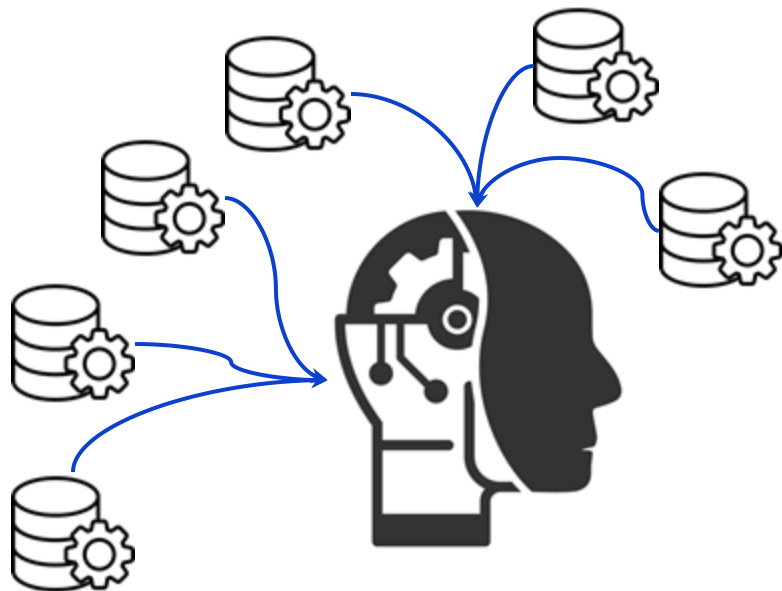
- 1) Given the success of OCEAN Assistant, there's an increasing need to include more data sources
- 2) RAG based on a single database of all data sources seem to be inefficient
- 3) Data retrieval pipeline becomes harder to maintain with increasing data sources

Our evaluation:

- The cost & effort of building a knowledge bot on one piece of data source is quite low
- User would prefer to have one interface to ask any question, rather than go to multiple platforms to ask domain-specific questions

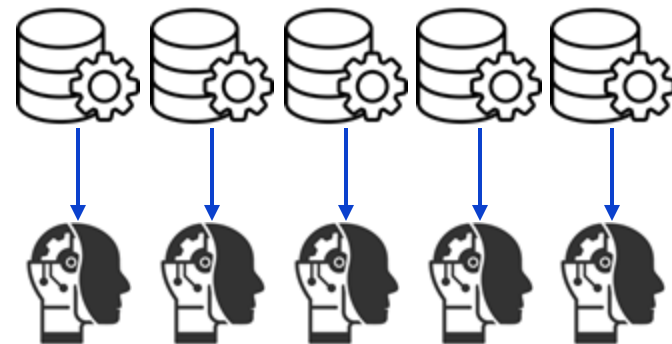
Reflection on Knowledge Retrieval Tasks

With ever-increasing data sources, is our approach scalable?



VS

Are these good enough?



Reflection on GenAI Coding Assistance

How does NExpert compare to Copilot, Cursor, Codeium, etc?

Key observations:

- 1) From a user perspective, the best place to provide coding assistance is right within an IDE
- 2) Pharmaverse packages are relatively new & considered niche
- 3) R language, generally speaking, is not as widely used when compared to python, javascript, etc.

Our evaluation:

- Public coding assistance tools (based on LLM only) have advantage in 1)
- NExpert (based on LLM + curated domain knowledge) has advantage in 2)
- Both solutions leverage backbone LLM to get R knowledge, NExpert may be slightly better given it has more R-related domain knowledge

Reflection on GenAI Coding Assistance

How does NExpert compare to Copilot, Cursor, Codeium, etc?

Can we cover all three?



Integrated with IDE



Knowledge on R language



NExpert



Knowledge on {pharmaverse}

Reflection on E2E Workflow

We built 2 chatbots for 2 business challenges, but we have way more than 2 challenges in our entire workflow

Key observations:

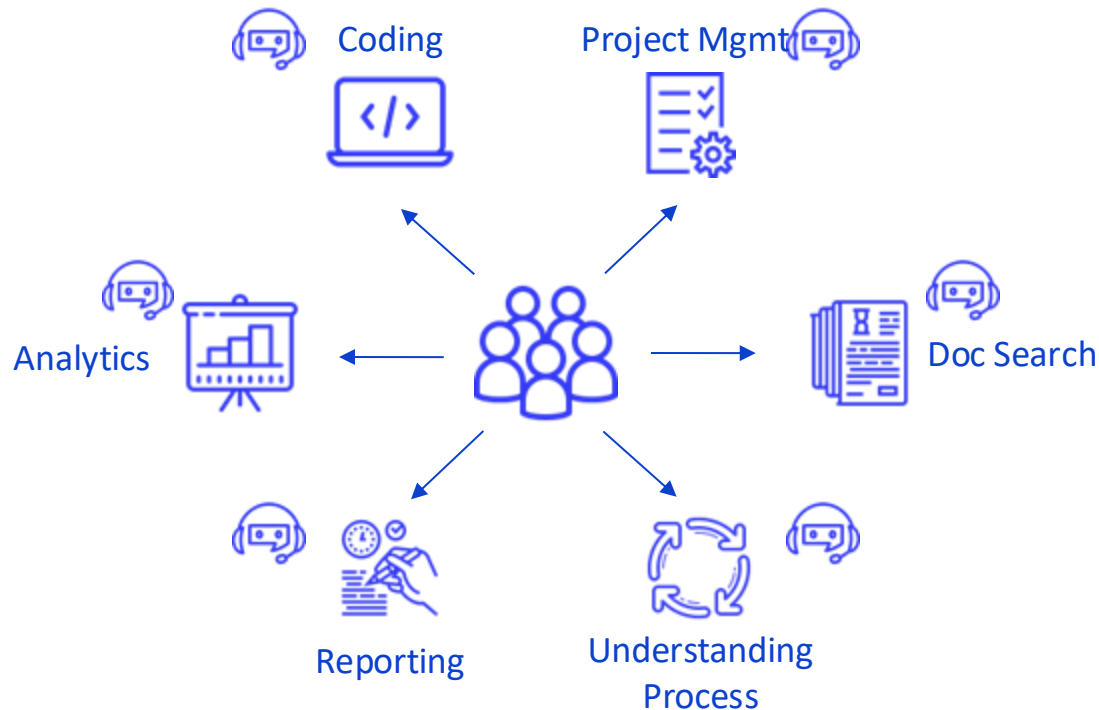
- 1) OCEAN Assistant & NExpert only aim to solve two business challenges: efficient knowledge retrieval & coding assistance, while the insights generation workflow is a complex process containing many more components than just coding and finding how-tos
- 2) We haven't leveraged some key advantages of LLM in our two use cases, for example, document comparison & writing, which are essential in many steps of our workflow

Our evaluation:

- More GenAI use cases are likely needed to bring more efficiency to our workflow
- Some steps in our workflow are considered routine but tedious, where a well-established automation pipeline could already bring much more efficiency

Reflection on E2E Workflow

We built 2 chatbots for 2 business challenges, but we have way more than 2 challenges in our entire workflow



Is this how we envision our data scientists work in the future?

Vision on GenAI for Insights Generation

Where should we focus on now & what can we dream for 2030?

Key Areas to Invest Now

How to get the best out of GenAI/LLM?

Data & Domain Knowledge

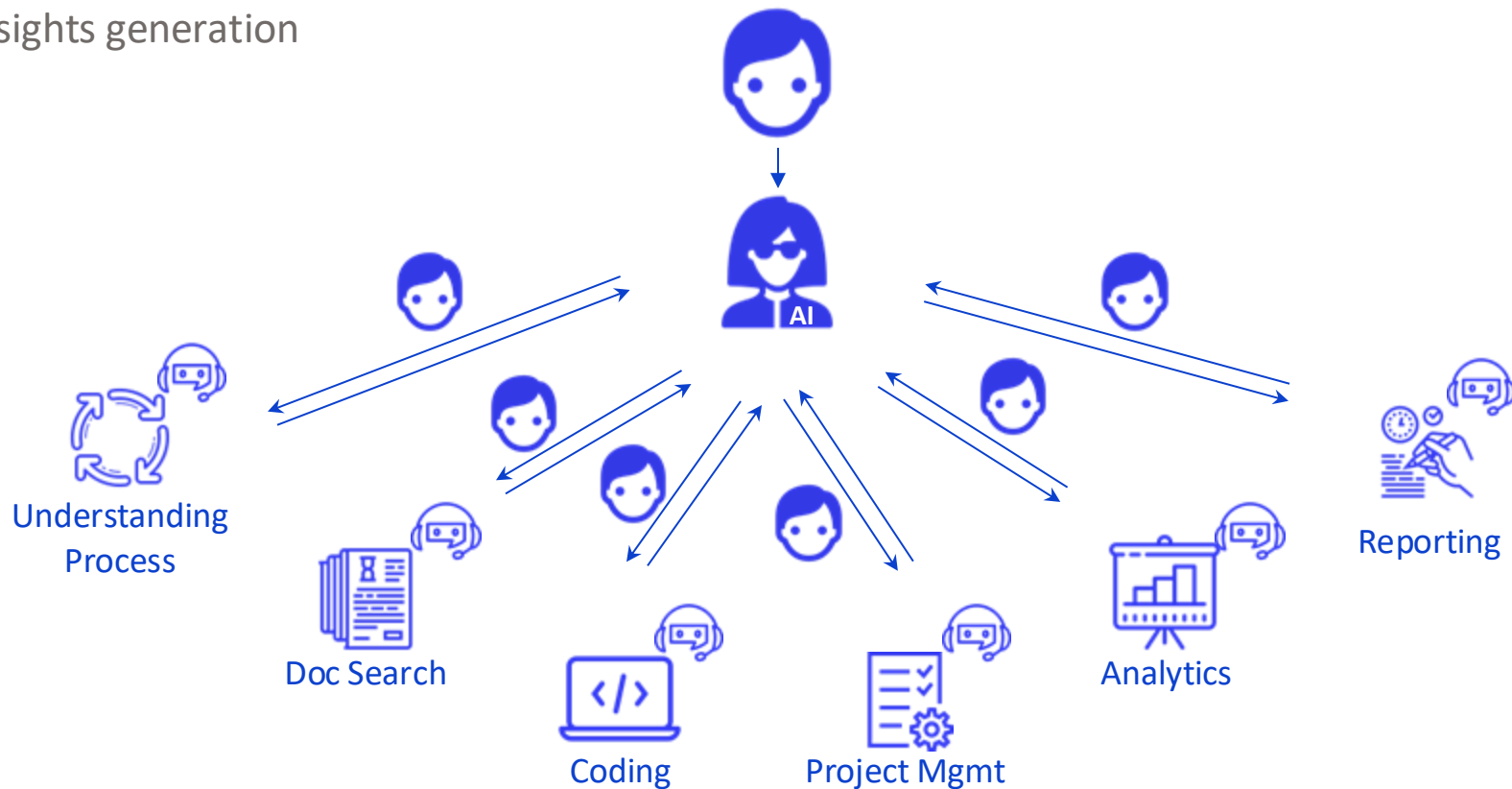
- **Well-designed database for all data sources & metadata related to insights generation**
 - *You have to know where to find the information yourself before building an AI to do it faster than you*
- **Well-maintained domain knowledge base**
 - *LLM is like a high-performance GPS, it will confidently guide you in the wrong direction if your map is outdated, incomplete, or full of errors*

Agentic Approach

- As insights generation workflow is a complex process containing many components, it becomes intuitive to consider leveraging an agentic approach, where an **AI Agent** breaks down a complex task into smaller sub-tasks and calls different **specialist AI Bots** to complete each task
- Specialist AI bot could work on complicated tasks like writing code, or simple tasks like triggering an automation pipeline

Future Outlook

A human-driven AI Agent-enabled workflow for faster and smoother clinical insights generation



GenAI in 2030

- Clinical Insights Generation Workflow completely transformed
- AI Agents oversee and manage the entire process
- Human Data Scientists act as reviewers and approvers
- Less time spent on repetitive tasks; More time on complex problem solving and strategic thinking



Q & A