phuse·share
connect·share·advance

Single Day Event

How to use RAG to build clinical trial knowledge base

Haiping Yu,
Jemincare

Disclaimer

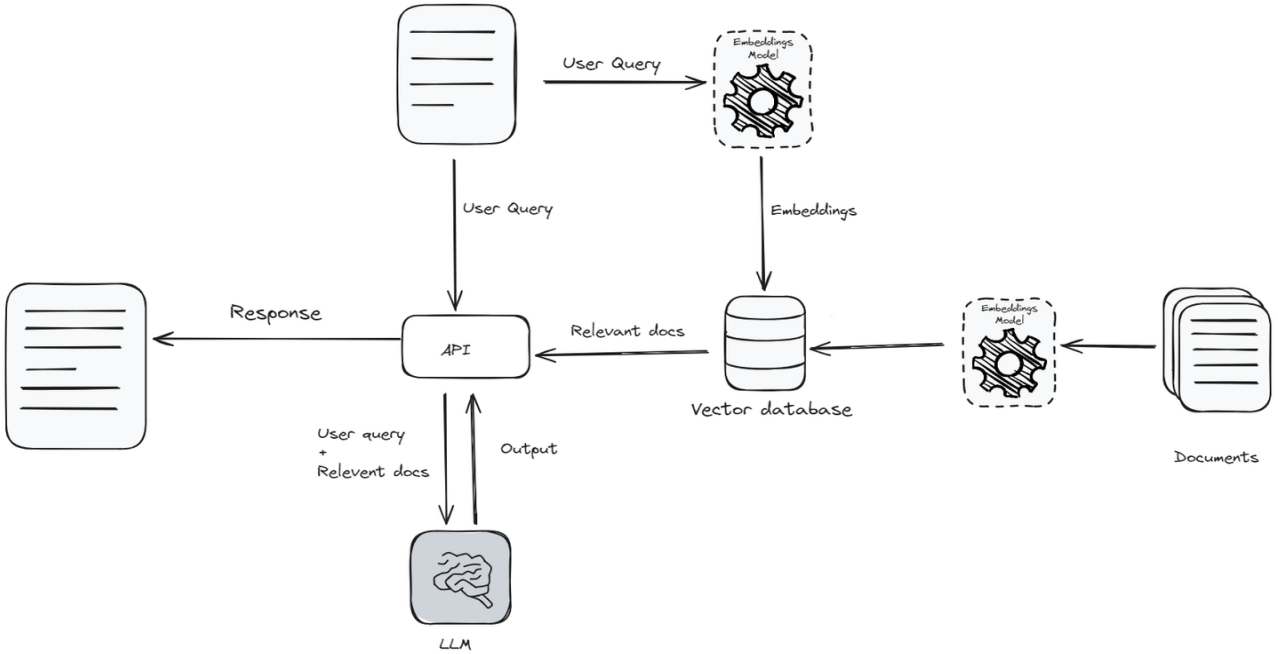
- The views and opinions expressed in this presentation are those of the author and do not necessarily reflect the official policy or position of Jemincare or any of its officers.

- ✓ What is RAG
- ✓ RAG vs. Long Context
- ✓ What is GraphRAG
- ✓ Build a clinical trial knowledge base

What is RAG?

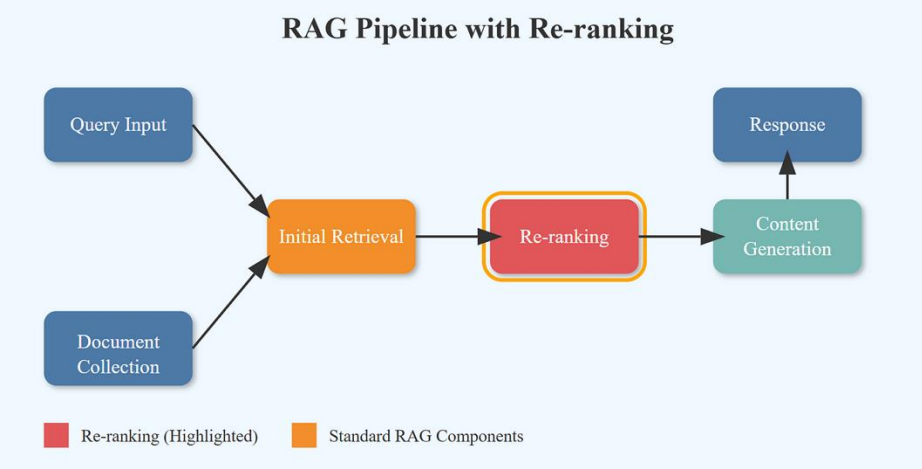
- RAG stands for Retrieval-Augmented Generation
- RAG combines retrieval and generation processes to enhance the capabilities of LLMs
- In RAG , the model retrieval information from a knowledge base or external sources
- RAG enables LLMs to produce higher-quality and more context-aware outputs

RAG Architecture



RAG Component

- External data
- Retriever
- Ranker
- Generator



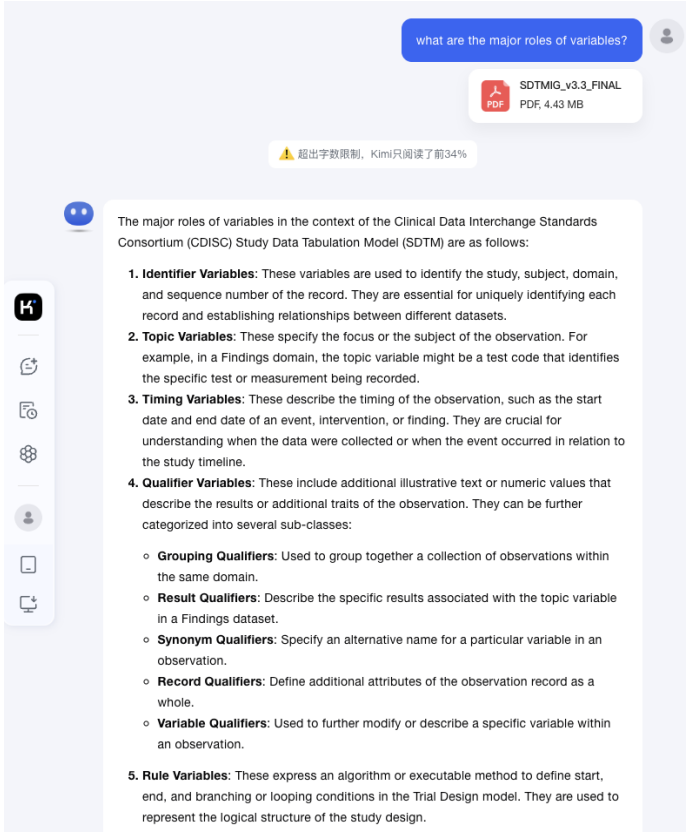
RAG Benefit

- Enhanced relevance
- Data privacy
- Up-to-date information
- Cost effective

RAG vs. Long Context

- Long Context is up to millions of tokens
- RAG and Long Context will co-exist for a long time

Long Context



2 Fundamentals of the SDTM

2.1 Observations and Variables

The SDTMIG for Human Clinical Trials is based on the SDTM's general framework for organizing clinical trials information that is to be submitted to regulatory authorities. The SDTM is built around the concept of observations collected about subjects who participated in a clinical study. Each observation can be described by a series of variables, corresponding to a row in a dataset. Each variable can be classified according to its Role. A Role determines the type of information conveyed by the variable about each distinct observation and how it can be used. Variables can be classified into five major roles:

- *Identifier* variables, such as those that identify the study, subject, domain, and sequence number of the record
- *Topic* variables, which specify the focus of the observation (such as the name of a lab test)
- *Timing* variables, which describe the timing of the observation (such as start date and end date)
- *Qualifier* variables, which include additional illustrative text or numeric values that describe the results or additional traits of the observation (such as units or descriptive adjectives)
- *Rule* variables, which express an algorithm or executable method to define start, end, and branching or looping conditions in the Trial Design model

The set of Qualifier variables can be further categorized into five sub-classes:

- *Grouping Qualifiers* are used to group together a collection of observations within the same domain. Examples include --CAT and --SCAT.
- *Result Qualifiers* describe the specific results associated with the topic variable in a Findings dataset. They answer the question raised by the topic variable. Result Qualifiers are --ORRES, --STRESC, and -- STRESN.
- *Synonym Qualifiers* specify an alternative name for a particular variable in an observation. Examples include --MODIFY and --DECOD, which are equivalent terms for a --TRT or --TERM topic variable, and
- --TEST and --LOINC, which are equivalent terms for a --TESTCD.
- *Record Qualifiers* define additional attributes of the observation record as a whole (rather than describing a particular variable within a record). Examples include --REASND, AESLIFE, and all other SAE flag variables in the AE domain; AGE, SEX, and RACE in the DM domain; and --BLFL, --POS, --LOC, -- SPEC and --NAM in a Findings domain
- *Variable Qualifiers* are used to further modify or describe a specific variable within an observation and are only meaningful in the context of the variable they qualify. Examples include --ORRESU, --ORNRHI, and --ORNRLO, all of which are Variable Qualifiers of --ORRES; and --DOSU, which is a Variable Qualifier of --DOSE.

RAG

Hi! I'm your assistant, what can I do for you?

what are the major roles of variables?

The major roles of variables in the provided knowledge base can be categorized as follows:

1. **Identifier**: Variables that uniquely identify entities such as studies, subjects, or records within a domain.
   - Examples: `STUDYID`, `DOMAIN`, `USUBJID`, `IESEQ`, `IESPID`
2. **Topic**: Variables that describe the main subject or content of the data.
   - Examples: `IETESTCD`, `IETEST`, `MIDSTYPE`
3. **Synonym Qualifier**: Variables that provide additional descriptive text related to the topic.
   - Example: `IETEST` (Inclusion/Exclusion Criterion)
4. **Grouping Qualifier**: Variables used for categorizing or grouping related records or criteria.
   - Examples: `IECAT`, `IESCAT`
5. **Rule**: Variables that express criteria in a form that can be executed by a computer, often desc ribing conditions under which something is planned to happen.
   - Example: `TIRL` (Inclusion/Exclusion Criterion Rule)
6. **Record Qualifier**: Variables that provide additional information about the record itself, such as version numbers or repetition indicators.
   - Examples: `TIVERS` (Protocol Criteria Versions), `TMRPT` (Disease Milestone Repetition In dicator)
7. **Result Qualifier**: Variables that capture the result of a criterion evaluation in either original or standard format.
   - Examples: `IEORRES` (I/E Criterion Original Result), `IESTRESC` (I/E Criterion Result in Std Format)

SDTMIG_v3.3_FINAL.pdf

2 Fundamentals of the SDTM

2.1 Observations and Variables

The SDTMIG for Human Clinical Trials is based on the SDTM's general framework for organizing clinical trials information that is to be submitted to regulatory authorities. The SDTM is built around the concept of observations collected about subjects who participated in a clinical study. Each observation can be described by a series of variables, corresponding to a row in a dataset. Each variable can be classified according to its Role. A Role determines the type of information conveyed by the variable about each distinct observation and how it can be used. Variables can be classified into five major roles:

- *Identifier* variables, such as those that identify the study, subject, domain, and sequence number of the record
- *Topic* variables, which specify the focus of the observation (such as the name of a lab test)
- *Timing* variables, which describe the timing of the observation (such as start date and end date)
- *Qualifier* variables, which include additional illustrative text or numeric values that describe the results or additional traits of the observation (such as units or descriptive adjectives)
- *Rule* variables, which express an algorithm or executable method to define start, end, and branching or looping conditions in the Trial Design model

The set of Qualifier variables can be further categorized into five sub-classes:

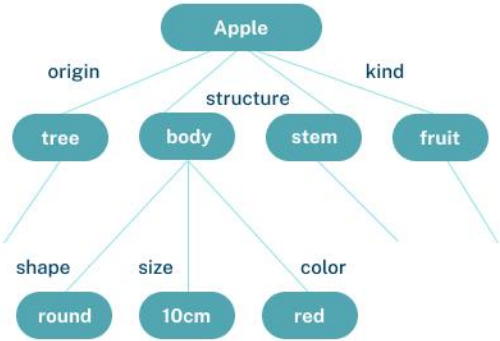
- *Grouping Qualifiers* are used to group together a collection of observations within the same domain. Examples include --CAT and --SCAT.
- *Result Qualifiers* describe the specific results associated with the topic variable in a Findings dataset. They answer the question raised by the topic variable. Result Qualifiers are --ORRES, --STRESC, and -- STRESN.
- *Synonym Qualifiers* specify an alternative name for a particular variable in an observation. Examples include --MODIFY and --DECOD, which are equivalent terms for a --TRT or --TERM topic variable, and
- --TEST and --LOINC, which are equivalent terms for a --TESTCD.
- *Record Qualifiers* define additional attributes of the observation record as a whole (rather than describing a particular variable within a record). Examples include --REASND, AESLIFE, and all other SAE flag variables in the AE domain; AGE, SEX, and RACE in the DM domain; and --BLFL, --POS, --LOC, -- SPEC and --NAM in a Findings domain
- *Variable Qualifiers* are used to further modify or describe a specific variable within an observation and are only meaningful in the context of the variable they qualify. Examples include --ORRESU, --ORNRHI, and --ORNRLO, all of which are Variable Qualifiers of --ORRES; and --DOSU, which is a Variable Qualifier of --DOSE.

Graph

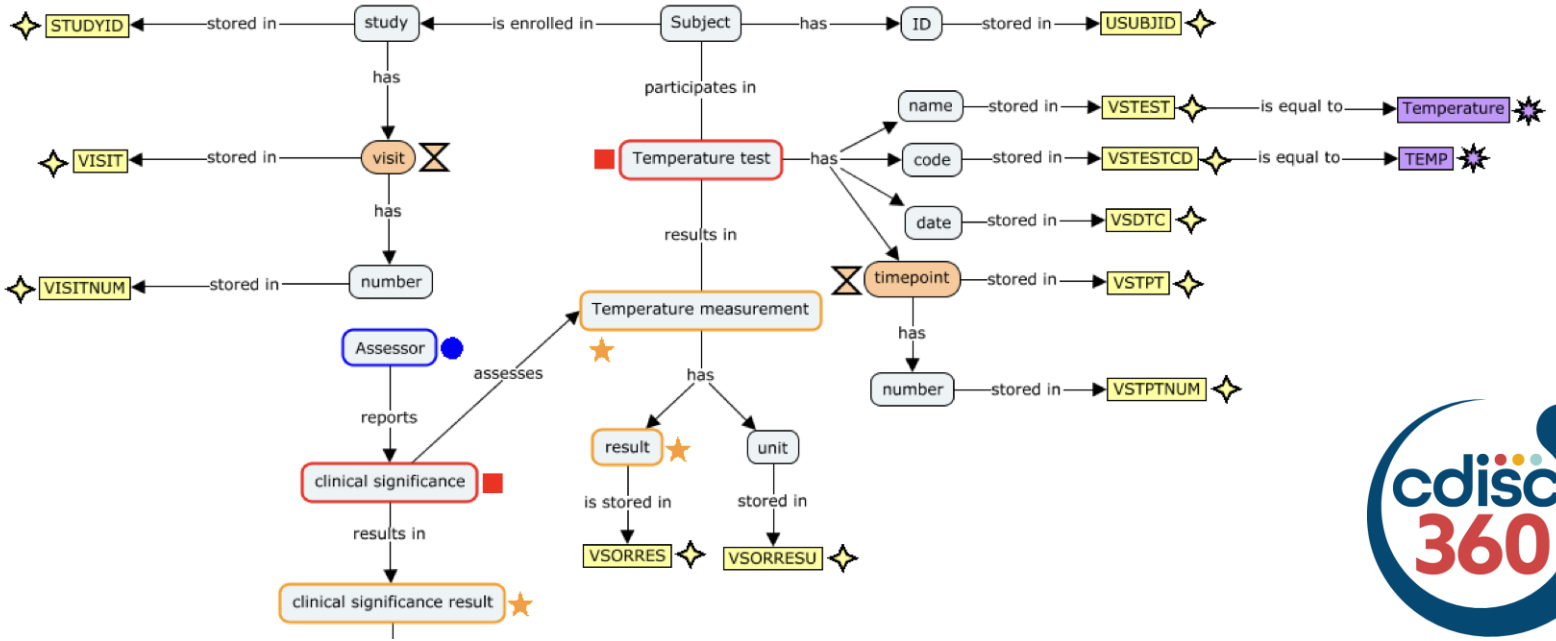
Human View of an Apple

Vector View of an Apple

Knowledge Graph View of an Apple



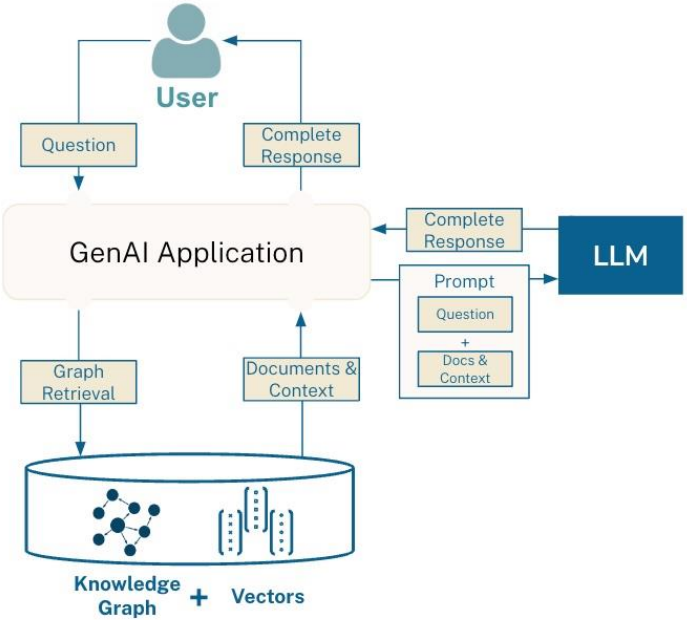
Knowledge Graph



Knowledge Graph vs LLMs

	Pros	Cons
Knowledge Graph	• Structural knowledge • Accuracy • Domain-specific knowledge • New knowledge	• incompleteness • Lacking language understanding
LLMs	• General knowledge • Language processing	• Implicit knowledge • Hallucination • Black-box • Lacking domain-specific /new knowledge

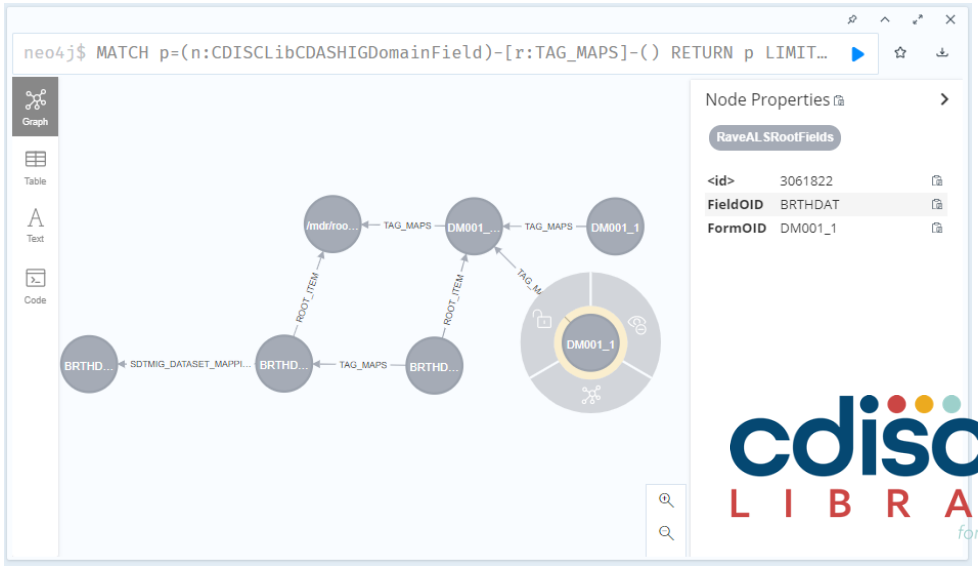
GraphRAG= Knowledge Graph + RAG

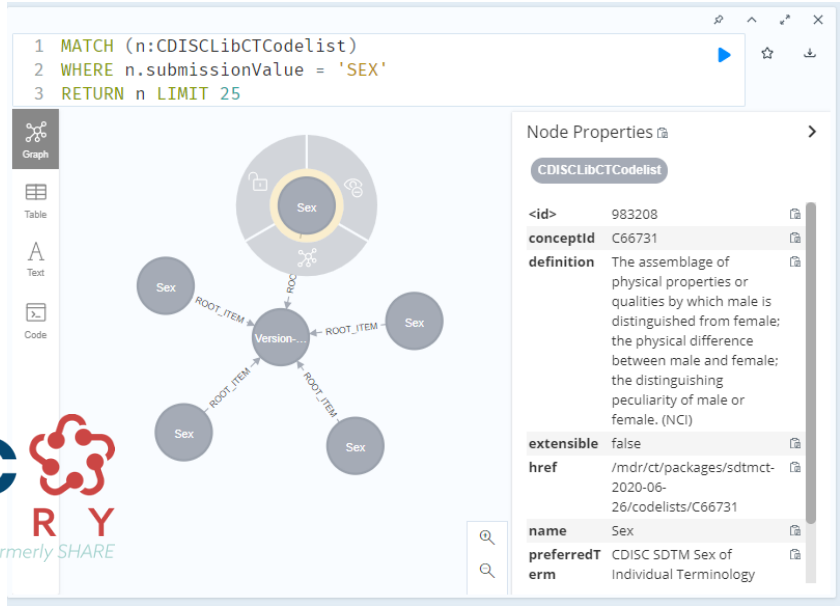


Clinical trial knowledge base (CTKB)

- RAG framework
  - RAGFlow- https://github.com/infiniflow/ragflow
- Graph database
  - neo4j
- LLM for embedding, re-ranking and chat
  - ZHIPU AI etc,

CTKB Graph database





cdisc LIBRARY
formerly SHARE

CTKB Architecture