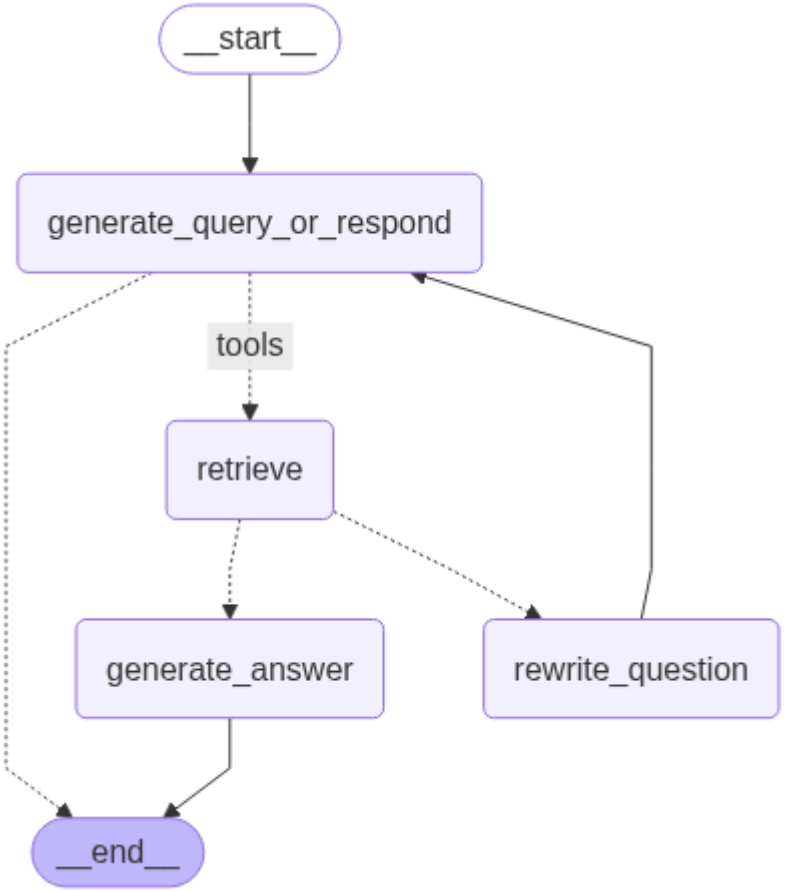
RAG

Agentic RAG

- Use an agent to determine whether to respond directly to the user or retrieve documents and then generate answers.

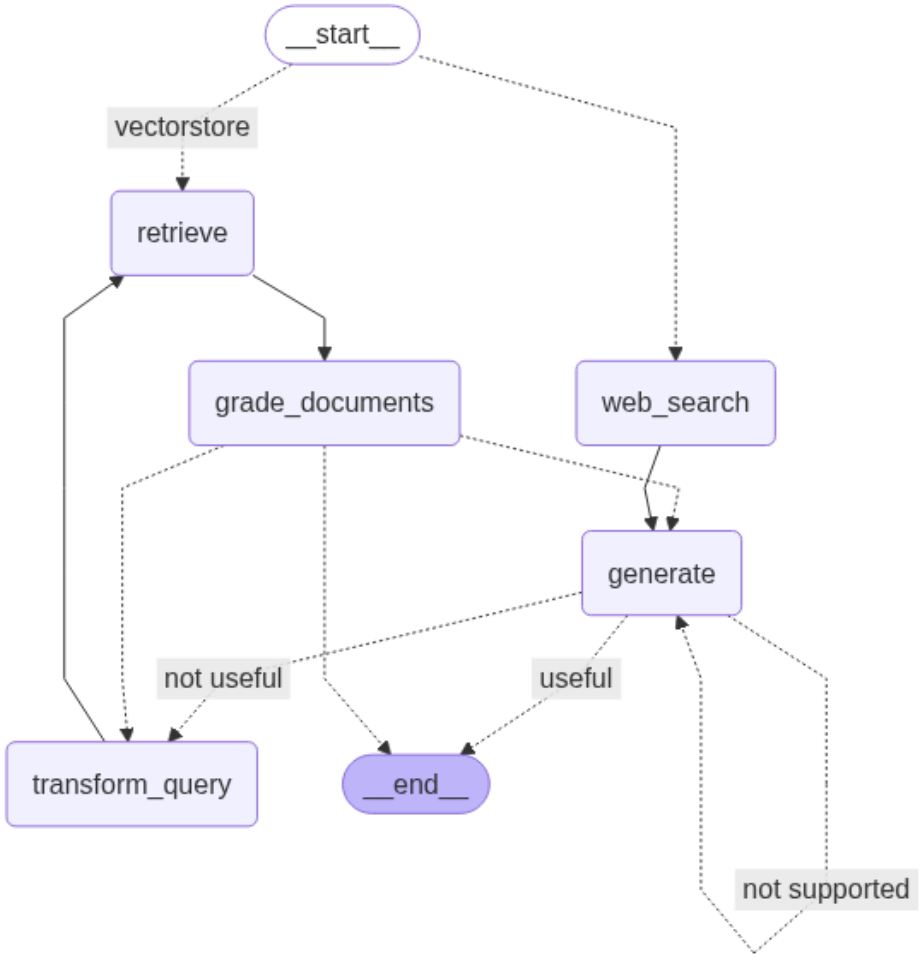


Coding:
Document preparation – Create retriever tool – Create LLM and bind tool – Grader – Rewrite question – Answer generation

Retrieve or not:
may skip retrieval and respond directly

Adaptive RAG

- Use route to determine which data source should be leveraged for answer generation, either document index or web search or any custom option
- Train a classifier to pre-determine complexity level of query and then dynamically adjust query handling strategy.
- Paper described how to construct training datasets to train classifier.



Coding:
Document preparation – Build index for document –- Build router for different source (retriever, web search & other custom option) – retrieval outcome grader + hallucination grader + answer grader – Rewrite question – Answer generation

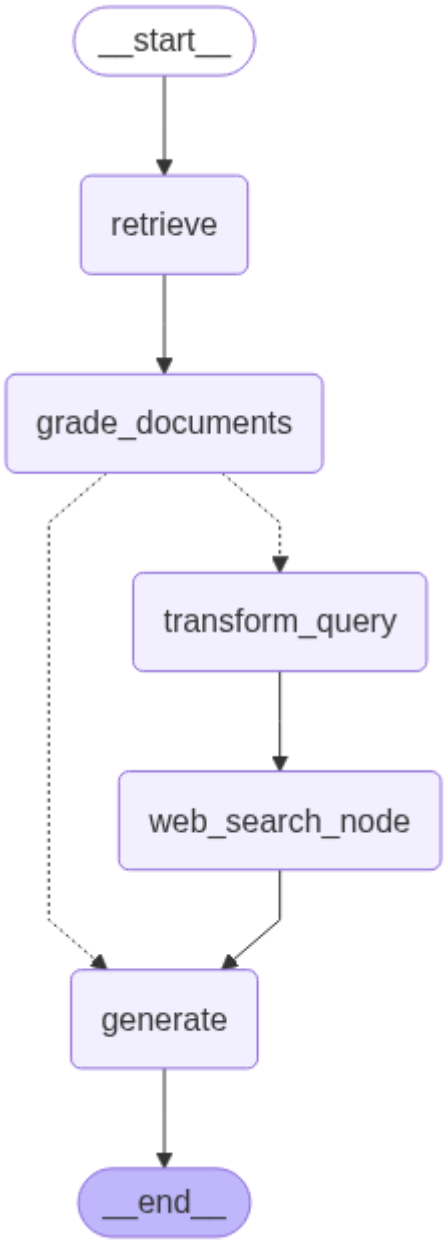
Advantage:

- More effective answer than a single step approach.
- Less computational costs than a multi-step approach.

Retrieve or not:
may skip retrieval and do web search

Corrective RAG

- Always do retrieval then self correct via grading plus web search



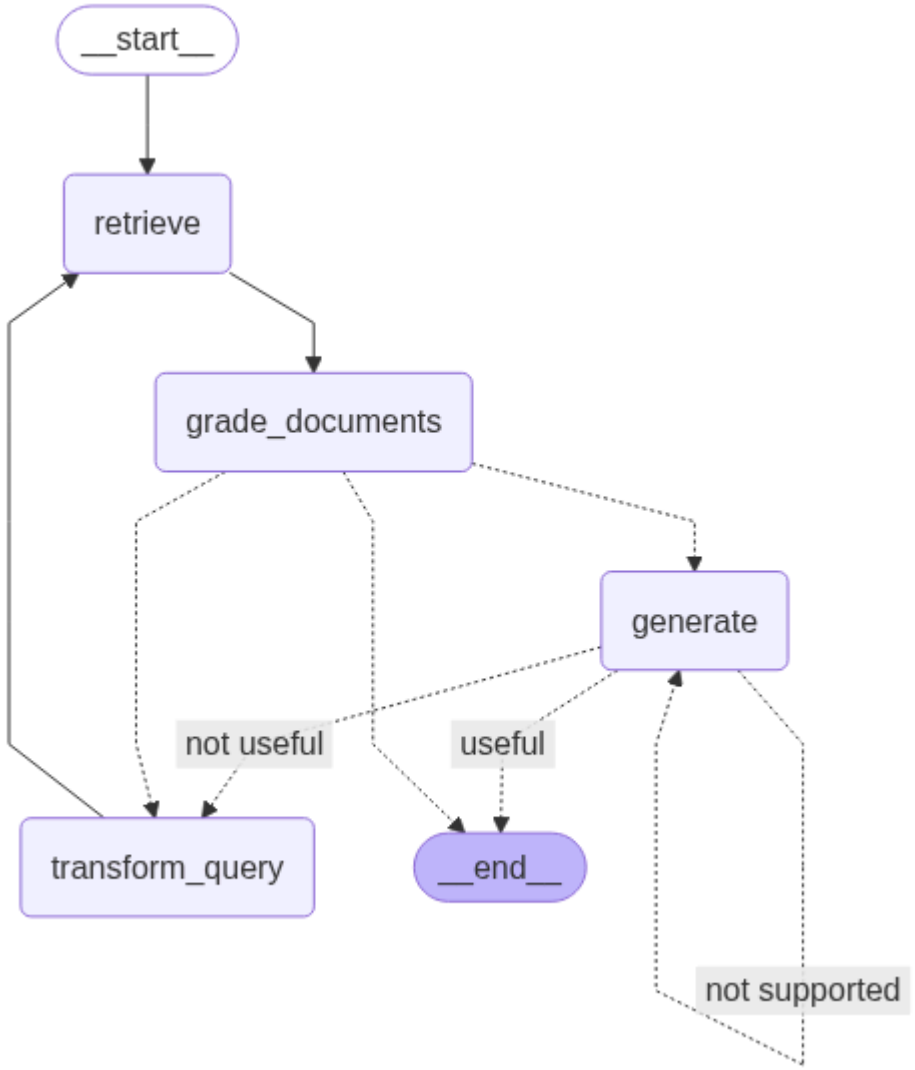
Coding:
Document preparation – Create retrieval tool and do retrieval – retrieval outcome(retrieved documents) grader – Rewrite question – seek other source eg. web search – Answer generation

Retrieve or not:
always do retrieval first

Advantage:
Use a retrieval evaluator to determine the accuracy of retrieval outcome
If retrieval is not relevant then self correct and seek other sources to answer questions.

Self RAG

- incorporate self reflection on retrieved documents compared to agentic RAG
- not include web search compared to adaptive RAG



Coding:
Document preparation – Create retrieval tool and do retrieval – retrieval outcome(retrieved documents) grader + hallucination grader + answer grader – Rewrite question – Answer generation