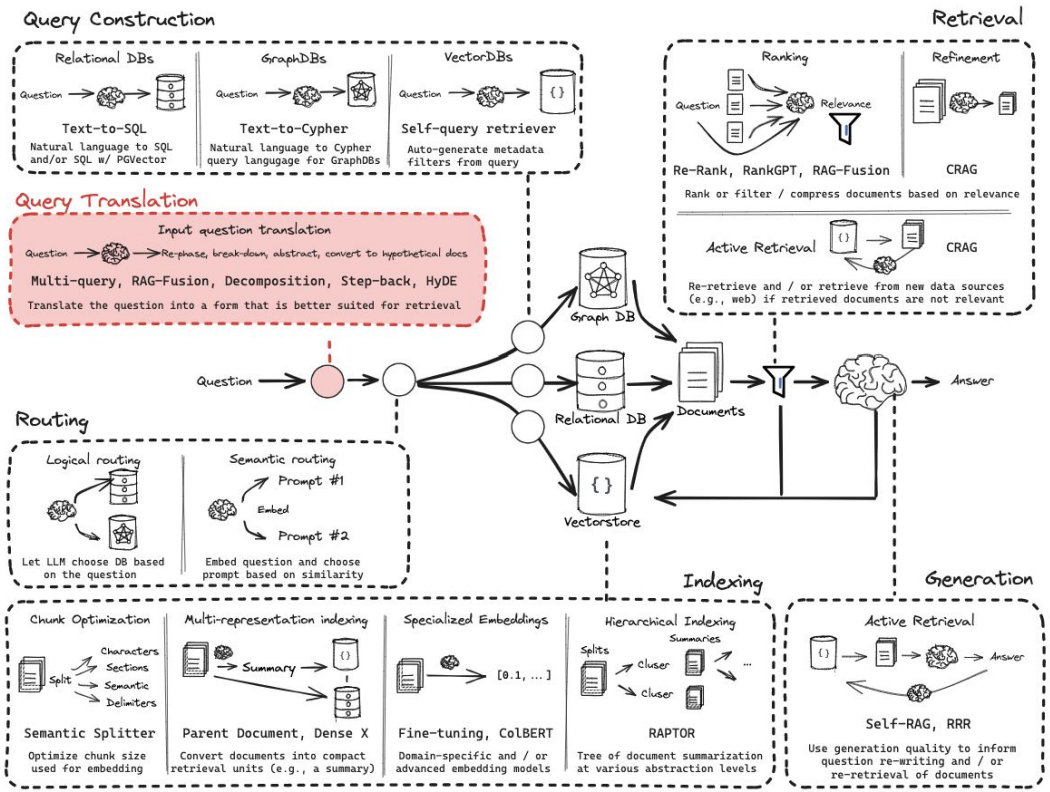
LangChain

RAG from scratch: Query Translation (Decomposition)

Lance Martin
Software Engineer, LangChain
@RLanceMartin

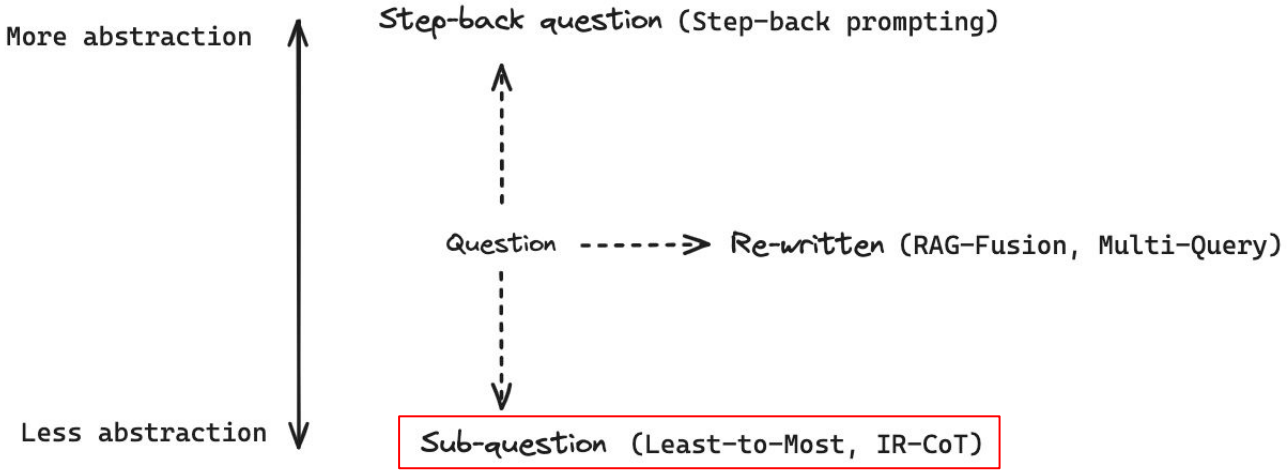
LangChain

Query Translation



LangChain

General approaches to transform questions



https://arxiv.org/pdf/2205.10625.pdf

LangChain

Least-to-most: Decompose problem into sub-problems, solve sequentially

(1) Decompose the problem

Q: “think, machine, learning”
A: “think”, “think, machine”, “think, machine, learning”

Table 1: Least-to-most prompt context (decomposition) for the last-letter-concatenation task. It can decompose arbitrary long lists into sequential subsists with an accuracy of 100%.

Q: “think, machine”
A: The last letter of “think” is “k”. The last letter of “machine” is “e”. Concatenating “k”, “e” leads to “ke”. So, “think, machine” outputs “ke”.

Q: “think, machine, learning”
A: “think, machine” outputs “ke”. The last letter of “learning” is “g”. Concatenating “ke”, “g” leads to “keg”. So, “think, machine, learning” outputs “keg”.

(2) Solve each step using prior sub-solutions

Table 2: Least-to-most prompt context (solution) for the last-letter-concatenation task. The two exemplars in this prompt actually demonstrate a base case and a recursive step.

https://arxiv.org/pdf/2205.10625.pdf

LangChain

IR-CoT: Interleave retrieval with CoT

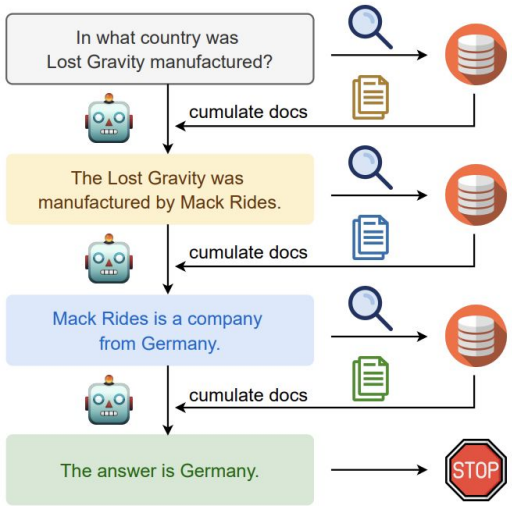
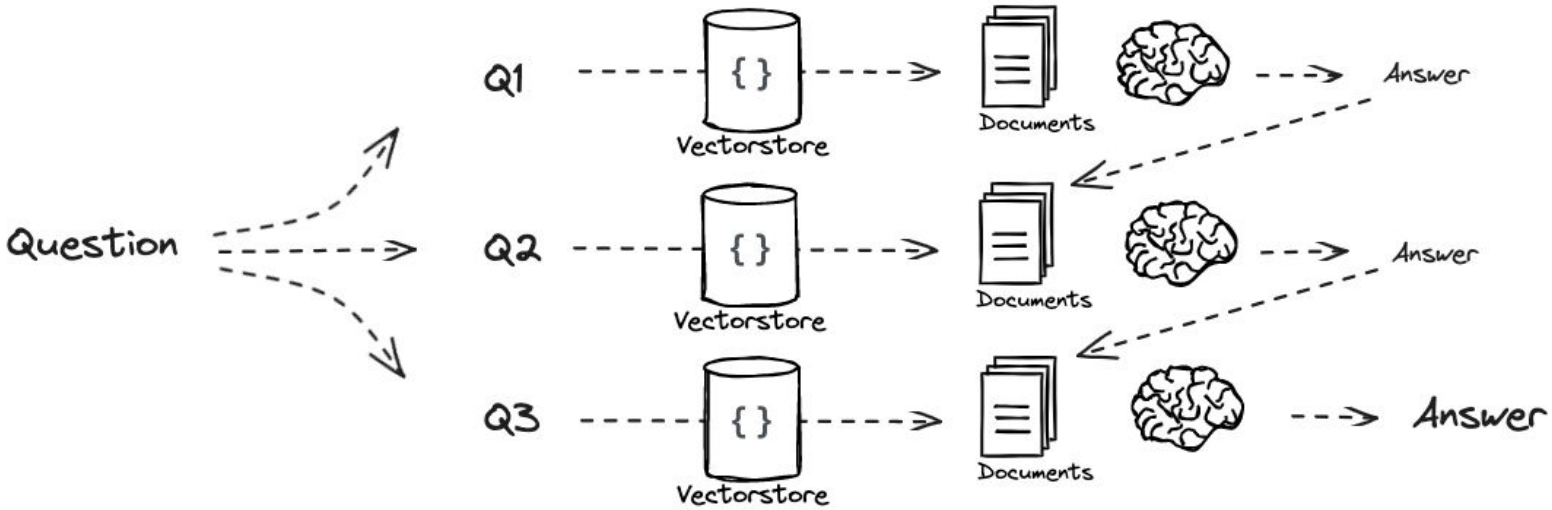


Figure 1: IRCoT interleaves chain-of-thought (CoT) generation and knowledge retrieval steps in order to guide the retrieval by CoT and vice-versa. This interleaving allows retrieving more relevant information for later reasoning steps, compared to standard retrieval using solely the question as the query.

https://arxiv.org/pdf/2212.10509.pdf

LangChain

Combine ideas: Dynamically retrieve to aid in solving the subproblems



Code walk-through