LLM Query Generation in RAG (Simple Explanation with Examples)

---

1. What is RAG?

RAG stands for Retrieval-Augmented Generation. It is a method where an model (like ChatGPT or Gemini) uses outside information to give better and more accurate answers.

Example:
If you ask 'What are the symptoms of tomato leaf blight?' — the AI might not remember the latest data. So, it first searches in a database or document collection for the answer, retrieves that information, and then uses it to generate a final, factual reply.

2. Why Query Generation is Needed

Before the model searches for the right information, it must understand what to look for. That's where 'query generation' comes in — the model converts the user's question into a form that can be used for searching.

Example:
User question: 'How can I stop white spots on my tomato leaves?'
Generated query: 'Tomato leaf white spot disease causes and prevention methods'.

3. Step-by-Step Process

Step 1 – User asks a question
Example: 'How do I fix yellow leaves in tomato plants?'

Step 2 – Clean and simplify the question
Remove unnecessary words or unclear parts.
Simplified: 'Fix yellow leaves tomato plants'

Step 3 – Generate a search query
Turn it into a search-friendly query like 'tomato leaf yellowing causes and treatments'.

Step 4 – Retrieve information
Use the generated query to find the most relevant text or data from the database or documents.

Step 5 – Choose useful parts
Pick 2–3 most related text chunks that can answer the question.

Step 6 – Create the final prompt
Combine the found information with the question. Example:

Context:
'Tomato leaves turn yellow due to nitrogen deficiency or overwatering.'

Question: 'How do I fix yellow leaves in tomato plants?'

Prompt to LLM: 'Using the context above, give a short and clear answer.'

Step 7 – LLM generates the answer
The model reads both the user question and the context to generate the final factual answer.

Output: 'Tomato leaves turn yellow mainly due to lack of nitrogen. Use a balanced fertilizer and avoid overwatering.'

4. Example Summary

Let's go through one full example:

User question: 'Why is my tomato plant having white spots on leaves?'

→ The system rewrites it to 'Tomato leaf white spots causes and treatment'.
→ It searches the document store and finds:
'White spots are often due to powdery mildew or fungal infection. Use fungicide and keep leaves dry.'
→ The LLM gets this information along with the question and gives the final answer:

'White spots on tomato leaves are caused by powdery mildew. Apply fungicide and ensure good air circulation.'

5. Why This Helps

This process helps the model give correct answers using up-to-date or specialized information instead of depending only on what it learned during training. It reduces wrong or made-up answers (hallucinations).

6. Simple Code Example

Here's a simple Python-style pseudocode showing how this works:

```
question = 'How to treat tomato leaf yellowing?'
clean_query = simplify(question)
results = vector_database.search(clean_query)
context = choose_best(results)
prompt = f'Context: {context}\nQuestion: {question}\nAnswer:'
answer = llm.generate(prompt)
print(answer)
```

7. Key Points to Remember

- RAG = Retrieve + Generate
- Query generation helps the system find the right data
- Chunking means splitting documents into small searchable parts
- LLM uses the retrieved context to give better, factual answers
- Examples make the process easier to understand and explain

This version is made simple and clear with examples for easier understanding of LLM query generation in RAG.