Iterative RAG

강사 장철원

Section 0

목차

Iterative RAG

Section 1. Iterative RAG

Section 1-1. Iterative RAG의 개념

Iterative RAG

- RAG를 통한 검색 결과가 충분하지 않으면?

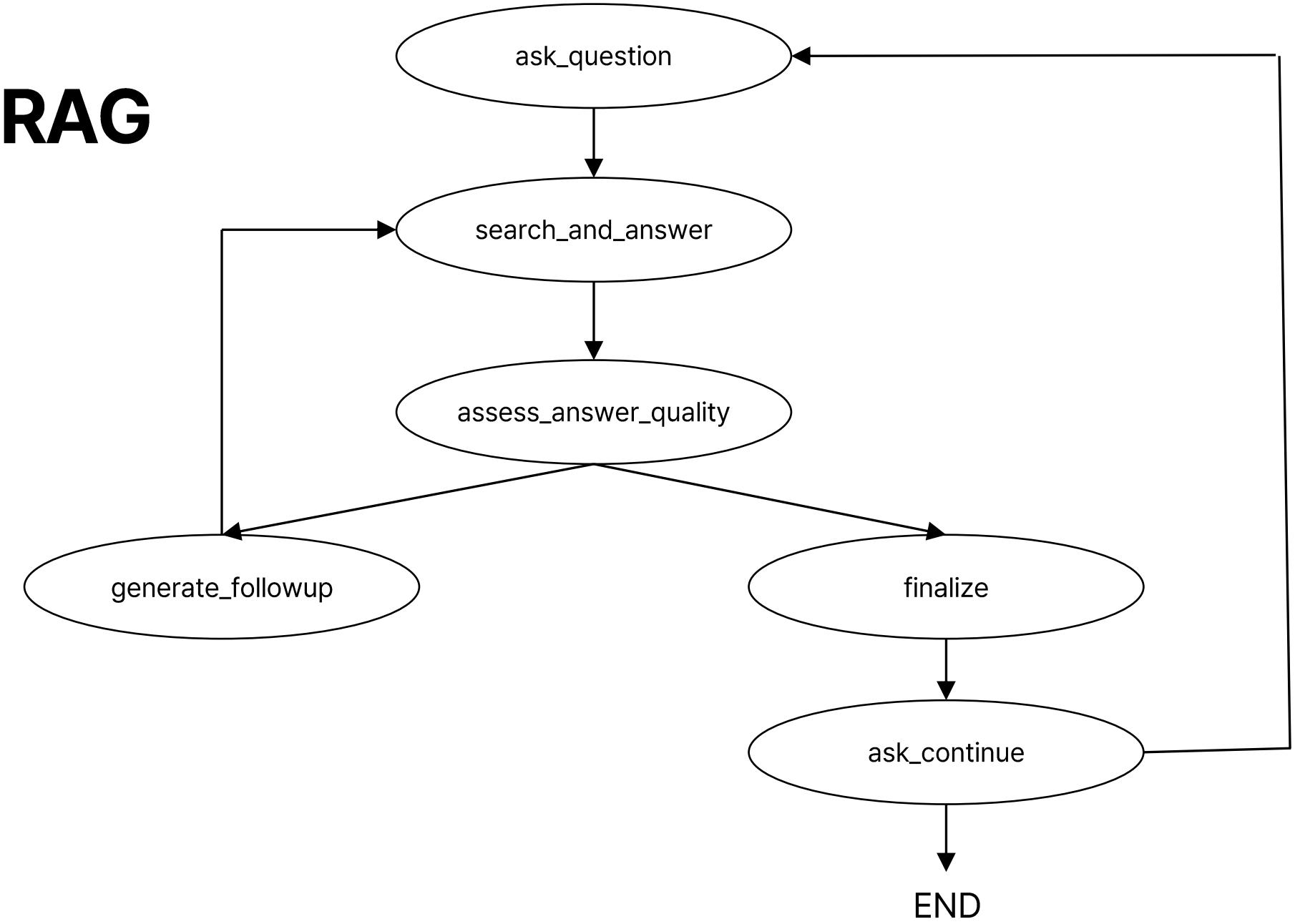
검색을 여러 번 반복하면서 더 정확하고 풍부한 답을 만드는 방식

Iterative RAG

Section 1. Iterative RAG

Section 1-2. Iterative RAG 실습

Iterative RAG



Iterative RAG

```python
import uuid
import pandas as pd
from langchain_chroma import Chroma
from langchain.schema import Document
from langgraph.graph import StateGraph, END
from langchain_huggingface import HuggingFaceEmbeddings, HuggingFacePipeline
from transformers import AutoTokenizer, AutoModelForQuestionAnswering, AutoModelForCausalLM, pipeline
from langchain_core.chat_history import InMemoryChatMessageHistory
from langchain_core.messages import get_buffer_string
from langchain_core.runnables import RunnableConfig
from langchain_core.chat_history import BaseChatMessageHistory
```

```python
# 1. 문서 불러오기
df = pd.read_csv('./data/data.csv', encoding='utf-8')
```

```python
# 2. 문서 리스트로 변환
texts = df["text"].tolist()
docs = [Document(page_content=text) for text in texts]
```

```python
# 3. 임베딩 모델
embedding_model = HuggingFaceEmbeddings(model_name="sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2")
```

```python
# 4. Chroma에 저장
vectorstore = Chroma.from_documents(
    documents=docs,
    embedding=embedding_model,
    persist_directory="./chromaDB"
)
```

Section

Iterative RAG 실습

Iterative RAG

```
# 5. 벡터 DB
retriever = Chroma(
    persist_directory="./chromaDB",
    embedding_function=embedding_model
).as_retriever(search_kwargs={"k": 3})
```

```
# 6. QA 모델 & 파이프라인
qa_model_id = "monologg/koelectra-base-v3-finetuned-korquad"
qa_tokenizer = AutoTokenizer.from_pretrained(qa_model_id)
qa_model = AutoModelForQuestionAnswering.from_pretrained(qa_model_id)

qa_pipeline = pipeline(
    "question-answering",
    model=qa_model,
    tokenizer=qa_tokenizer,
    device=-1)

qa_llm = HuggingFacePipeline(pipeline=qa_pipeline)
```

```
Device set to use cpu
```

```
# 7. 재질문 생성 모델
rewrite_model_id = "skt/kogpt2-base-v2"
rewrite_tokenizer = AutoTokenizer.from_pretrained(rewrite_model_id)
rewrite_model = AutoModelForCausalLM.from_pretrained(rewrite_model_id)
rewrite_tokenizer.model_max_length = 1024

rewrite_pipeline = pipeline(
    "text-generation",
    model=rewrite_model,
    tokenizer=rewrite_tokenizer,
    max_new_tokens=64
)

rewrite_llm = HuggingFacePipeline(pipeline=rewrite_pipeline)
```

```
Device set to use cpu
```

Iterative RAG

```python
# 8. 세션 관리
chats_by_session_id = {}
def get_chat_history(session_id: str) -> BaseChatMessageHistory:
    if session_id not in chats_by_session_id:
        chats_by_session_id[session_id] = InMemoryChatMessageHistory()
    return chats_by_session_id[session_id]
```

Section

Iterative RAG 실습

노드 정의(1)

```python
# 9. 노드 정의
def ask_question(state, config: RunnableConfig):
    session_id = config["configurable"]["session_id"]
    history = get_chat_history(session_id)
    question = input("질문을 입력해주세요: ").strip()
    history.add_user_message(question)
    return {
        "question": question,
        "original_question": question,
        "attempt": 1,
        "history": history
    }

def search_and_answer(state):
    question = state["question"]
    docs = retriever.invoke(question)
    context = "\n".join([doc.page_content for doc in docs])
    result = qa_pipeline(question=question, context=context)
    return {**state, "docs": docs, "context": context, "answer": result["answer"]}

def assess_answer_quality(state):
    answer = state["answer"].strip()
    attempt = state.get("attempt", 1)
    if len(answer) < 10 and attempt < 3:
        print("답변이 불충분하여 추가 질의가 필요합니다.")
        return {**state, "needs_followup": True}
    print(f"\n질문: {state['question']}\n답변: {answer}")
    return {**state, "needs_followup": False}
```

Section

Iterative RAG 실습

노드 정의(2)

```python
def generate_followup(state):
    original_question = state["original_question"]
    prompt = f"원 질문: {original_question}\n더 구체적으로 묻기 위해 재질문을 만들어주세요."
    followup = rewrite_pipeline(prompt)[0]["generated_text"]
    print(f"\n재질문 생성: {followup.strip()}")
    return {
        **state,
        "question": followup.strip(),
        "attempt": state["attempt"] + 1
    }

def finalize(state):
    session_id = state.get("session_id")
    state["history"].add_ai_message(state["answer"])
    return state

def ask_continue(state):
    cont = input("\n계속하시겠습니까? (예/아니오): ").strip()
    state["continue"] = cont.startswith("예")
    return state

def should_followup(state):
    return "generate_followup" if state.get("needs_followup") else "finalize"

def should_continue(state):
    return "ask_question" if state.get("continue") else END
```

Section

Iterative RAG 실습

LangGraph 구성

```python
# 10. LangGraph 구성
graph = StateGraph(dict)
graph.add_node("ask_question", ask_question)
graph.add_node("search_and_answer", search_and_answer)
graph.add_node("assess_answer_quality", assess_answer_quality)
graph.add_node("generate_followup", generate_followup)
graph.add_node("finalize", finalize)
graph.add_node("ask_continue", ask_continue)

graph.set_entry_point("ask_question")
graph.add_edge("ask_question", "search_and_answer")
graph.add_edge("search_and_answer", "assess_answer_quality")
graph.add_conditional_edges("assess_answer_quality", should_followup, {
    "generate_followup": "generate_followup",
    "finalize": "finalize"
})
graph.add_edge("generate_followup", "search_and_answer")
graph.add_edge("finalize", "ask_continue")
graph.add_conditional_edges("ask_continue", should_continue, {
    "ask_question": "ask_question",
    END: END
})
```

```
<langgraph.graph.state.StateGraph at 0x1967f1bda50>
```

Section

Iterative RAG 실습

실행

```python
# 11. 실행
session_id = str(uuid.uuid4())
config = {"configurable": {"session_id": session_id}}
app = graph.compile()
app.invoke({}, config=config)
```

```
질문을 입력해주세요: 공정 조건 이상 발생 시 어떻게 처리하나요?
답변이 불충분하여 추가 질의가 필요합니다.

재질문 생성: 원 질문: 공정 조건 이상 발생 시 어떻게 처리하나요?
더 구체적으로 묻기 위해 재질문을 만들어주세요. 이번에 나오는 '생선살'은
냉장이 안 됩니다
매운탕이 맛있다.
그것도 한 숟가락에 쏙들어가서 먹고싶은 분들은 한 숟가락에 다 먹어줘야 하는듯 하더라구요.
오늘 이 가격에 살수 있다면!!!
이렇게 짬

질문: 원 질문: 공정 조건 이상 발생 시 어떻게 처리하나요?
더 구체적으로 묻기 위해 재질문을 만들어주세요. 이번에 나오는 '생선살'은
냉장이 안 됩니다
매운탕이 맛있다.
그것도 한 숟가락에 쏙들어가서 먹고싶은 분들은 한 숟가락에 다 먹어줘야 하는듯 하더라구요.
오늘 이 가격에 살수 있다면!!!
이렇게 짬
답변: Shift 리더에게 즉시 보고합니다

계속하시겠습니까? (예/아니오): 아니오
{'question': "원 질문: 공정 조건 이상 발생 시 어떻게 처리하나요?\n더 구체적으로 묻기 위해 재질
이 맛있다.\n그것도 한 숟가락에 쏙들어가서 먹고싶은 분들은 한 숟가락에 다 먹어줘야 하는듯 하더라
 'original_question': '공정 조건 이상 발생 시 어떻게 처리하나요?',
 'attempt': 2,
 'history': InMemoryChatMessageHistory(messages=[HumanMessage(content='공정 조건 이상 발생
```

감사합니다.

Q & A