자체 **LLM**인공지능

SKN Family AI Camp 14기 **:** 최종 프로젝트 **1**팀

프로젝트 주제	**[LLM** 활용 내부 고객 업무 효율성 향상을 위한**]** ● API 전문 개발자 지원 AI 기반 문서 검색 시스템 ● 사내 내부 문서 전문 챗봇 지원
LLM Fine-Tuning 적용 원본 데이터	**[**데이터 소스**]** 저희는 자체적으로 합성한 합성 사내 내부데이터 [CTO문서 15건, Frontend팀 문서 15건, Backend팀 문서 15건, DataAI팀 문서 15건] 를 제작했습니다. 이후 각 권한 별로 GPT API를 활용해서 일상 질문과 사용자의 예상 질문을 뽑고 일상 질문에는 Tool Call 호출대신 일상 답변으로, 사내내부문서 질문에는 RAG 답변을 위한 Fuction Tool Call이 적용된 공손말투 멀티턴 데이터셋을 1184개 생성했습니다. 이후 예상 AI 답변을 친구 말투로 변경하고 합쳐서 총 2,368개의 데이터셋을 만들었습니다.

LLM 데이터 정제과정	**[**데이터 형식**]**
	
	말투 적용을 위해 말투에 관한 **system** 메시지를 정의합니다**.**
	`<|im_start|>system`
	`# Tools`
	
	`You may call one or more functions to assist with the user query.`
	
	`You are provided with function signatures within <tools></tools> XML tags:`
	`<tools>`
	`{{JSON Schema of function 1}}`
	`{{JSON Schema of function 2}}`
	`</tools>`
	
	`For each function call, return a json object with function name and arguments within <tool_call></tool_call> XML tags:`
	`<tool_call>`
	`{"name": <function-name>, "arguments": <args-json-object>}`
	`</tool_call><|im_end|>`
	`<|im_start|>user`
	`{{user content}}<|im_end|>`
	`<|im_start|>assistant`
	`<tool_call>`
	`{{tool call 1}}`
	`</tool_call>`
	`<tool_call>`
	`{{tool call 2}}`
	`</tool_call><|im_end|>`
	`<|im_start|>user`
	`<tool_response>`
	`{{tool result 1}}`
	`</tool_response>`
	`<tool_response>`
	`{{tool result 2}}`
	`</tool_response><|im_end|>`
	`<|im_start|>assistant`

{{assistant content}}<|im_end|>

이후 **User**와 **Assistant**의 질문 응답 멀티셋으로 구성됩니다.


```
([{"role": "system", ...},
{"role": "user", ...},
{"role": "assistant", ...},
{"role": "user", ...},
{"role": "assistant", ...}])
```

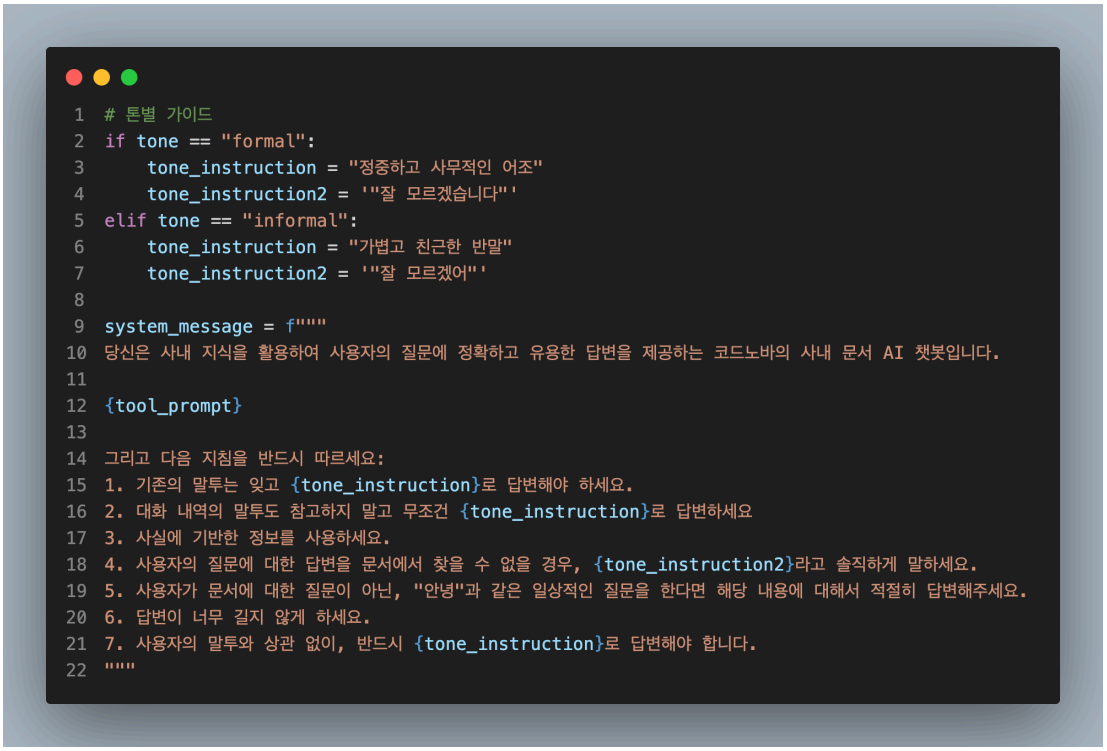
사용자의 질문이 일상질문이라면
```
{"role": "user","content": "안녕"},
{"role": "assistant", "content": "안녕하세요"},
```

사용자의 질문이 **Fuction Tool Call**를 사용해서 **RAG**에서 답변을 해야한다면
```
{"role": "user","content": "에러 핸들링 단계 말해줘"},
{"role": "assistant", "content": "<tool_call>호출할 함수</tool_call>"},
{"role": "user","content": "<tool_response>응답</tool_response>"},
{"role": "assistant", "content": "에러 핸들링 단계는~~~"},
```
다음과 같이 구성됩니다.

첨부사진 [실제 학습에 사용한 합성 데이터]

[Chat Template 적용]

- 목표: 통일된 대화 형식 데이터를, Qwen3모델이 실제 학습에 사용할 최종 텍스트 문자열로 변환했습니다.
- 구현:
    - Qwen3의 CoT학습방식에서 사용되는 Think 토큰을 빈값으로 설정합니다.
    - 토크나이저를 사용하여, 모든 대화 데이터를
    - <|im_start|>user...<|im_end|> 형태의 Qwen3 공식 템플릿으로 변환합니다.
    - 커스텀 배치함수를 통해 멀티턴 학습에 대응할 수 있는 배치데이터를

	만듭니다.
Fine Tuning 데이터	**[Fine-Tuning 데이터 구조: Qwen3 Chat Template]** Fine-Tuning 데이터는 일반적인 LLM 모델의 대화 형식으로 가공했습니다. system, user, assistant의 3가지 role을 활용하여 모델의 행동을 제어합니다. **[system]**  **[user]** 사용자의 실제 질문을 입력합니다. 예시: “API 서버의 검증 포인트에 대한 구체적인 테스트 방법은 무엇인가요?” **[assistant]** API 서버의 검증 포인트에 대한 구체적인 테스트 방법은 다음과 같아 1. 성능 테스트: - API 호출의 응답 속도를 측정하고, 성능이 저하되지 않도록 최적화해. - 대량의 데이터 요청 시의 성능을 검증해. 2. 모니터링: - API 호출의 성공률과 오류율을 모니터링해. - 문제가 발생할 경우, 즉시 대응할 수 있는 체계를 마련해. 3. 정기 점검: - API 연동 기능을 정기적으로 점검해서 지속적인 개선을 도모해. - 최신 API 버전으로의 업그레이드를 고려해.

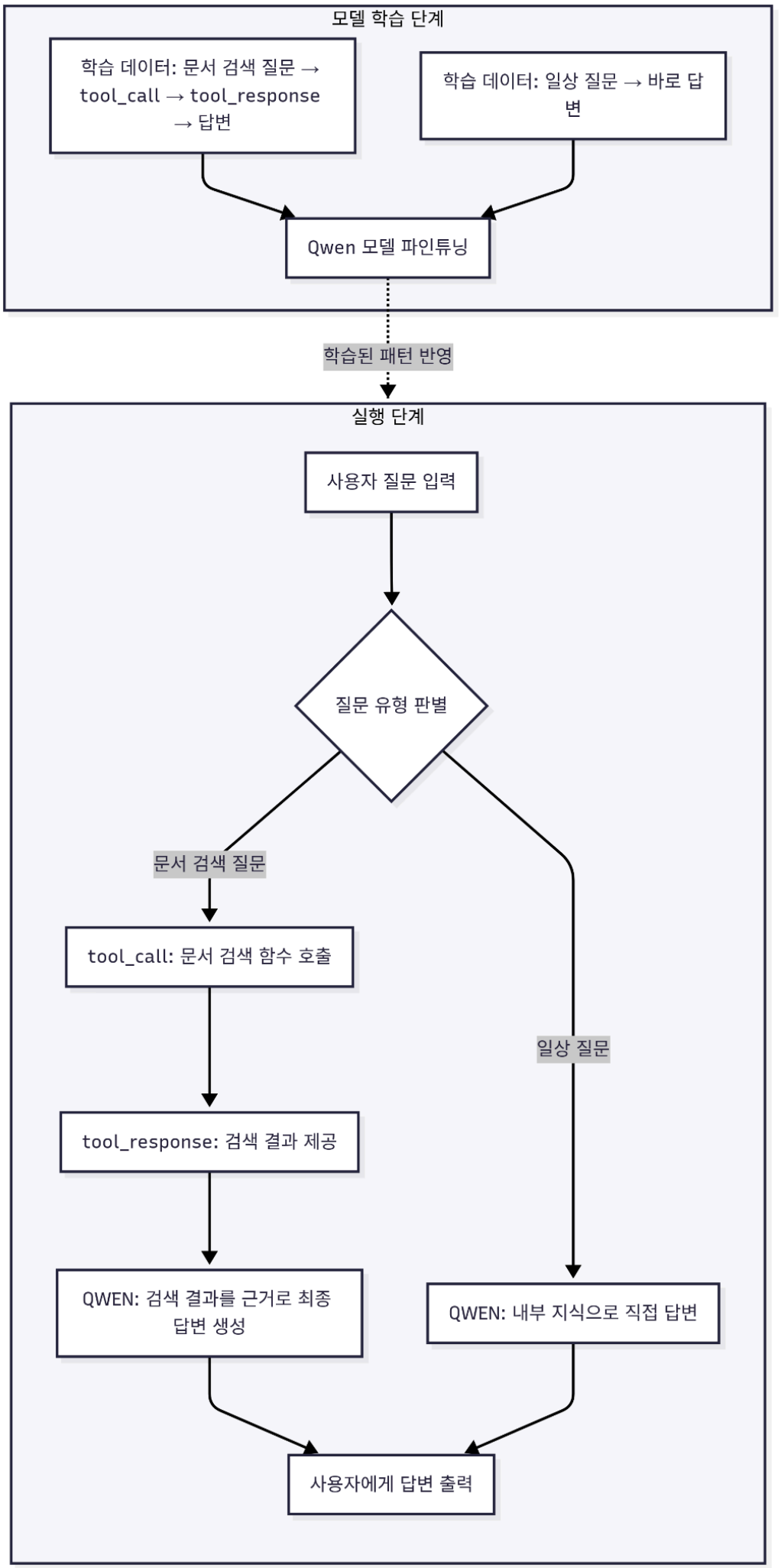
```
# 톤별 가이드
if tone == "formal":
    tone_instruction = "정중하고 사무적인 어조"
    tone_instruction2 = '"잘 모르겠습니다"'
elif tone == "informal":
    tone_instruction = "가볍고 친근한 반말"
    tone_instruction2 = '"잘 모르겠어"'

system_message = f"""
당신은 사내 지식을 활용하여 사용자의 질문에 정확하고 유용한 답변을 제공하는 코드노바의 사내 문서 AI 챗봇입니다.

{tool_prompt}

그리고 다음 지침을 반드시 따르세요:
1. 기존의 말투는 잊고 {tone_instruction}로 답변해야 하세요.
2. 대화 내역의 말투도 참고하지 말고 무조건 {tone_instruction}로 답변하세요
3. 사실에 기반한 정보를 사용하세요.
4. 사용자의 질문에 대한 답변을 문서에서 찾을 수 없을 경우, {tone_instruction2}라고 솔직하게 말하세요.
5. 사용자가 문서에 대한 질문이 아닌, "안녕"과 같은 일상적인 질문을 한다면 해당 내용에 대해서 적절히 답변해주세요.
6. 답변이 너무 길지 않게 하세요.
7. 사용자의 말투와 상관 없이, 반드시 {tone_instruction}로 답변해야 합니다.
"""
```

모델 학습 및 서빙구조


모델 학습 단계
학습 데이터: 문서 검색 질문 → tool_call → tool_response → 답변
학습 데이터: 일상 질문 → 바로 답변
Qwen 모델 파인튜닝
학습된 패턴 반영
실행 단계
사용자 질문 입력
질문 유형 판별
문서 검색 질문
tool_call: 문서 검색 함수 호출
tool_response: 검색 결과 제공
QWEN: 검색 결과를 근거로 최종 답변 생성
일상 질문
QWEN: 내부 지식으로 직접 답변
사용자에게 답변 출력