

# Google Gemini LLM

작성일: 2024-08-28

작성자: 유정연

## 1. 목적 및 요구사항

본 POC 문서는 2차 전처리 파이프라인 구성을 위해 사용할 LLM에 대한 정보를 기술할 목적으로 작성되었다.

본 프로젝트의 데이터 파이프라인에서 1차 전처리 단계는 특수 기호를 제거하는 등 불필요한 텍스트 데이터를 제거하는 것이 목적 이었다면, 2차 전처리 단계는 주어진 데이터에서 필요한 부분을 추출 및 요약하는 것이 주목적이다. 하지만 현 수집된 데이터는 형태가 일정하지 않은 비정형 텍스트 데이터로서 이를 정규표현식이나 패턴 분석 등 정형분석으로 접근하기에 어려움이 있다. 따라서 이를 LLM(Large Language Model)을 활용하여 텍스트 기반 텍스트 추출 및 요약 task를 진행하고자 한다. 해당 task의 목표 및 세부 요구사항은 다음과 같다.

목적	수집한 공고 데이터를 에서 누락되거나 추출하지 못한 데이터를 파생하는 것
요구사항	데이터에서 기술스택, 직무 요구사항, 우대사항 등 텍스트 추출 및 요약 가능
	빠른 LLM 응답속도
	적은 비용

## 2. Google Gemini란?

Google Gemini는 Google에서 제공하는 AI 모델 및 툴킷이다. 이는 자연어 처리 (NLP)와 관련된 다양한 작업을 지원하며, Google의 최신 AI 연구와 기술이 반영되어 있다. Gemini는 다양한 언어 생성 및 이해 작업을 위한 강력한 기능을 제공하며, AI 모델 개발 및 배포를 단순화하는 목적으로 설계되었다. 여러 모델이 있으며, 현 POC 문서에서는 두 가지 모델을 테스트 하였다. 작성일 기준 Gemini LLM은 무료로 제공 중이다.

- Gemini-1.5-Pro: 멀티모달 모델, 섹션당 2백만 입력 토큰 지원, 반응 속도는 빠르지 않음
- Gemini-1.5-flash: 텍스트 전용 모델, 섹션당 10만 입력 토큰 지원, 반응 속도 빠름

## 3. Gemini 개발 환경

Google Gemini는 Google AI Studio(<https://aistudio.google.com>)를 통해 접근할 수 있다. AI Studio는 클라우드 기반의 통합 개발 환경을 제공하는 Google AI 인터페이스이며, 데이터 분석, 모델 개발, 실험 관리, 배포 등을 한 곳에서 수행할 수 있도록 한다. 해당 사이트에서 prompt 테스트 후, 해당 prompt를 API code로 변환 가능하다.

만약 API를 Python 코드로 실행할 경우 아래 CLI 명령어로 pip 설치가 필요하다

```
$ pip install google-generativeai
```

## 4. Gemini API 예제

```
import google.generativeai as genai

genai.configure(api_key=gemini_api_key[ 'GEMINI_API' ])

# Create the model
generation_config = {
    "temperature": 0.7,
    "top_p": 0.95,
    "top_k": 64,
    "max_output_tokens": 8192,
    "response_mime_type": "text/plain",
}

model = genai.GenerativeModel(
    model_name="gemini-1.5-flash",
    generation_config=generation_config,
)

chat_session = model.start_chat(
    history=[
    ]
)

# 예제: 글 생성
prompt = "Translate the following English text into French: 'Hello, how are you?'"
response = chat_session.send_message(prompt )
print(response.text)

# 예제: 텍스트 분류
prompt = "A review of the latest smartphone model from XYZ company."
response = chat_session.send_message(prompt )
print(response.text)
```