Google Gemma 2

> 구글의 최신 오픈 소스 언어 모델인 **Gemma 2**에 대한 이 문서는 Gemma 2의 개요, 설치 및 사용 방법, 그리고 다양한 활용 예시에 대해 설명합니다.

1. 개요

Google Gemma 2는 구글이 Gemini 모델 개발에 사용한 동일한 연구와 기술을 바탕으로 제작된 경량형 대형 언어 모델(LLM) 제품군입니다. 주요 특징은 다음과 같습니다.

- **모델 크기 및 변형**

> Gemma 2는 다양한 용도와 하드웨어 환경에 맞춰 2B, 9B, 27B 등 여러 크기로 제공됩니다. 각 크기는 사전 훈련(Pre-trained) 모델과 지시 조정(Instruction-tuned) 모델로 구분되어, 사용자 요구에 맞게 선택할 수 있습니다.

- **아키텍처 및 설계**

  - 텍스트-투-텍스트, 디코더 전용 Transformer 아키텍처를 채택하여 다양한 텍스트 생성 작업(질의응답, 요약, 추론 등)에 적합합니다.
  - 상대적으로 작은 모델 크기에도 불구하고, 최신 기술(예: 로컬-글로벌 어텐션 교차, 그룹 쿼리 어텐션, 지식 증류 등)을 도입하여 성능을 극대화하였습니다.

- **학습 데이터**

> 웹 문서, 코드, 수학 등 다양한 소스의 방대한 텍스트 데이터를 기반으로 훈련되었으며, 27B 모델의 경우 최대 13조 개 이상의 토큰을 사용하여 학습되었습니다.

- **개방성과 활용성**

  - 모델 가중치가 공개되어 있어, 연구자와 개발자가 자유롭게 미세 조정(fine-tuning)하거나 자체 애플리케이션에 통합할 수 있습니다.
  - Keras 3.0, PyTorch, JAX, Hugging Face Transformers 등 여러 프레임워크를 통한 다중 플랫폼 지원을 제공합니다.

출처: Google Gemma Docs (core) | Gemma 2 model card

2. 설치 방법 및 사용법

> Gemma 2는 다양한 개발 환경에서 손쉽게 설치하고 사용할 수 있도록 여러 방식으로 제공됩니다.

2-1. Hugging Face Transformers 사용

> Hugging Face 라이브러리를 이용하여 Gemma 2 모델을 불러오고 텍스트를 생성할 수 있습니다.

```
from transformers import AutoTokenizer, AutoModelForCausalLM

# 모델 이름은 사용하려는 크기에 따라 선택합니다.
model_name = "google/gemma-2-27b-it"  # 또는 "google/gemma-2-9b-it"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)

prompt = "양자 얽힘(quantum entanglement)의 개념을 간단히 설명해줘."
inputs = tokenizer(prompt, return_tensors="pt")
outputs = model.generate(**inputs, max_length=200)
print(tokenizer.decode(outputs[0], skip_special_tokens=True))
```

2-2. Keras를 이용한 사용

> TensorFlow/Keras 환경에서도 Gemma 2를 손쉽게 사용할 수 있습니다. KerasNLP 라이브러리를 통해 모델을 불러오는 예제입니다.

```
import tensorflow as tf
from keras_nlp.models import GemmaCausalLM

# Gemma 2의 2B 영어 사전 훈련 모델 불러오기 (예시)
model = GemmaCausalLM.from_preset("gemma_2b_en")

prompt = "양자 얽힘의 개념을 간단히 설명해줘."
output = model.generate(prompt, max_length=200)
print(output)
```

2-3. 기타 개발 환경

- **Colab/Kaggle Notebook**: 구글에서 제공하는 Colab이나 Kaggle 환경을 통해 쉽게 Gemma 2 모델에 접근하고 미세 조정할 수 있습니다.
- **분산 학습 및 미세 조정**: Keras, JAX, PyTorch 등의 프레임워크를 활용하여 대규모 모델의 분산 튜닝(distributed tuning)도 지원합니다.

> **참고:** Gemma 2는 다양한 프레임워크와의 호환성을 제공하므로, 사용 환경에 맞게 적절한 라이브러리와 설정을 선택하면 됩니다.

3. 활용 예시

> Gemma 2의 강력한 텍스트 생성 능력과 효율성을 활용하면 다양한 애플리케이션에 적용할 수 있습니다.

3-1. 챗봇 및 대화형 인터페이스

- **고객 서비스 챗봇**: 사용자의 질문에 신속하고 정확한 답변을 제공하는 챗봇을 구축할 수 있습니다.
- **역사/교육용 챗봇**: 특정 도메인(예: 역사, 과학) 지식을 바탕으로 대화형 인터페이스를 구현하여 사용자와 상호작용할 수 있습니다.

예시 코드 (체인릿(Chainlit)과 Ollama를 활용한 챗봇 구축 예제):

```
from langchain_community.llms import Ollama
from langchain.prompts import ChatPromptTemplate
import chainlit as cl

@cl.on_chat_start
async def on_chat_start():
    await cl.Message(content="안녕하세요, 저는 Gemma입니다. 무엇을 도와드릴까요?").send()
    model = Ollama(model="gemma2")
    prompt = ChatPromptTemplate.from_messages([
        ("system", "당신은 역사 전문가입니다. 정확하고 친절한 답변을 제공합니다."),
        ("human", "{question}")
    ])
    runnable = prompt | model
    cl.user_session.set("runnable", runnable)

@cl.on_message
async def on_message(message: cl.Message):
    runnable = cl.user_session.get("runnable")
    async for chunk in runnable.astream({"question": message.content}):
        await cl.Message(content=chunk).send()
```

3-2. 문서 요약 및 질의응답 시스템

- **문서 요약**: 긴 문서를 간결하게 요약하거나, 특정 주제에 대한 정보를 추출할 수 있습니다.
- **질의응답(QA) 시스템**: 사용자 입력에 기반해 관련 정보를 검색·생성하는 RAG(검색 증강 생성) 시스템을 구축할 수 있습니다.

3-3. 코드 생성 및 코딩 어시스턴트

- **코드 자동 완성 및 생성**: 프로그래밍 문제나 코드 작성 시, Gemma 2의 언어 이해 능력을 활용하여 코드 자동 완성 기능을 구현할 수 있습니다.
- **개발자 도우미**: 코드 리뷰, 오류 수정, 문서 작성 등 개발자 도구로 활용할 수 있습니다.

3-4. 창의적 콘텐츠 생성

- **마케팅 콘텐츠**: 광고 카피, 소셜 미디어 게시글, 블로그 포스트 등 다양한 창작 텍스트를 자동 생성합니다.
- **창작 소설 및 시**: 창의적인 문학 작품이나 시를 생성하는 데에도 사용할 수 있습니다.

> **출처:** Google Gemma Docs (core) | Gemma 2 model card

결론

Google Gemma 2는 최신 연구 기술이 반영된 경량형 대형 언어 모델로, 뛰어난 성능과 다양한 크기, 그리고 다중 프레임워크 지원을 통해 연구자와 개발자 모두에게 강력한 텍스트 생성 및 처리 도구를 제공합니다. Hugging Face, Keras, Colab 등 다양한 환경에서 쉽게 설치하고 사용할 수 있으며, 챗봇, 질의응답, 코드 생성, 문서 요약 등 광범위한 응용 분야에 활용 가능합니다.