

# LLM(Large Language Model)을 사용한 AI 챗봇 연구



201724416 김대영

201824408 강주호

201824579 정영진

지도교수 김호원 교수님

---

## 목 차

1. 서론.....	1
1.1. 연구 배경.....	1
1.2. 기존 문제점.....	2
1.3. 연구 목표.....	3
2. 연구 배경.....	4
2.1. 데이터 수집.....	4
2.2. 배경 지식.....	5
2.3. 시스템 구성.....	8
3. 연구 내용.....	9
3.1. 데이터 전처리.....	9
3.2. 모델 선정.....	11
3.3. 모델 학습.....	12
3.3. 답변 생성.....	14
3.4. 프론트 및 백엔드.....	16
3.5. API.....	18
4. 연구 결과 분석 및 평가.....	20
5. 결론 및 향후 연구 방향.....	23
6. 구성원별 역할 및 개발 일정.....	24
7. 참고 문헌.....	25

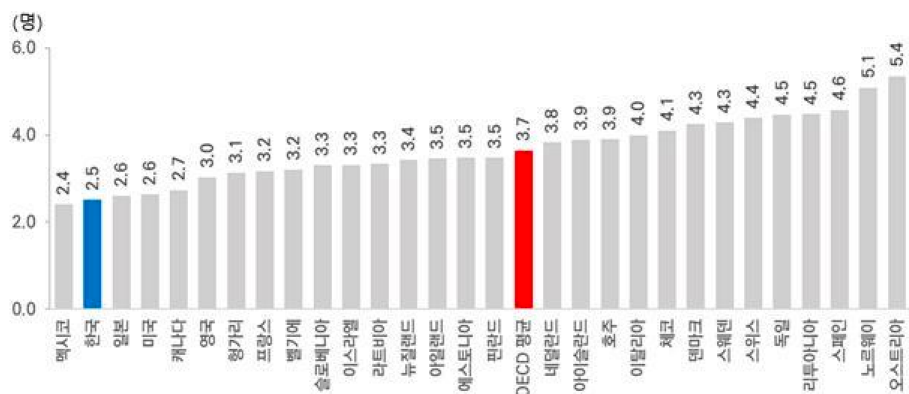
# 1. 서론

## 1.1. 연구 배경

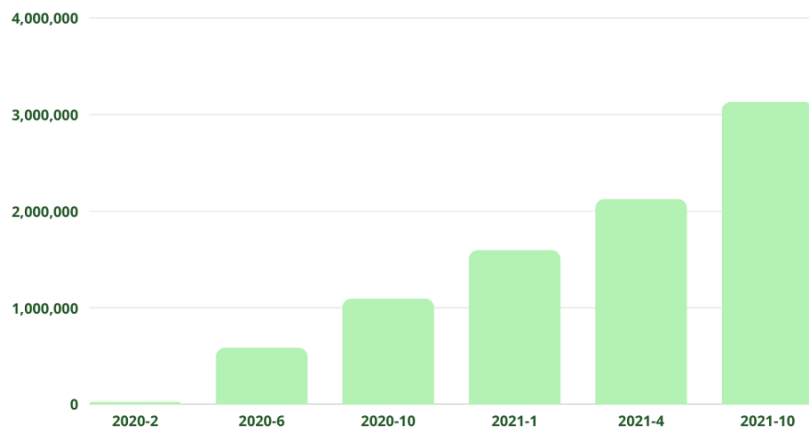
한국은 인구의 고령화 문제와 함께 의료 분야에서 인력 부족과 치료 비용 상승과 같은 문제에 직면하고 있다. 보건 복지부의 연구에 따르면, 한국의 인구 1000명 당 의사 수는 2.6명으로, OECD 평균인 3.7명보다 약 30% 부족한 것으로 나타났다. 이는 OECD 회원국 중 최하위인 모습이다.

covid-19 팬데믹은 의료 수요의 급증과 더불어 비대면 진료의 중요성을 더욱 부각시켰다. 이 상황에서 의료 분야에서 고려되는 해결책 중 하나는 '의료용 챗봇'이다. 의료용 챗봇은 시간 제약 없이 환자들이 정보나 조언을 얻을 수 있으며 의료 비용을 절감하고 다양한 이점을 제공한다.

그러나, 현재의 의료 챗봇은 정확성 문제 등으로 실질적인 활용에 제약이 있다. 이러한 상황을 고려하여, 더 효과적이고 현실적인 도움을 제공할 수 있는 의료용 챗봇을 개발하는 것이 목표이다.



인구 1000명 당 의사수 (출처: 보건복지부)



비대면 진료 환자 수 추이 (출처:보건복지부)

## 1.2. 기존 문제점

### ● 정확성의 문제

현재 시장에 제공되는 의료용 AI 챗봇의 정확성에는 문제가 있다. 현재 제공 중인 의료용 AI 챗봇은 주로 룰 기반 방식(Rule-based)을 사용하며, 이로 인해 특정 질문에 대한 응답 패턴을 명시적으로 정의할 수 있는 장점이 있다. 그러나 이러한 방식은 모든 상황을 예측하기 어렵고, 새로운 정보나 상황에 대한 업데이트가 수동적으로 이루어져야 한다. 따라서 이미 정의된 규칙 이외의 질문에 정확한 답변을 제공하기 어렵다는 문제가 있다.

### ● 개인화와 인간적 요소의 부재

또한, 현재 의료용 AI 챗봇은 개인화된 의료 조언과 응답을 제공하는 데 어려움이 있다. 의료는 증상 뿐만 아니라 개인의 상황, 의견, 감정 등 다양한 인간적인 요소에 의존하며, 기존의 모델로는 이러한 요소를 충분히 학습하고 대응하기 어렵다. 결과적으로 환자는 상황에 따라 고려 받지 못하고, 비슷한 증상을 가진 환자들에게 유사한 진단이 내려져 환자 입장에서 제대로 된 의료 서비스를 받지 못하는 불편함을 느낄 수 있다.

---

### 1.3. 연구 목표

LLM (Large Language Model)을 활용하여 의료용 AI 챗봇을 개발하고, 이를 통해 사용자들에게 보다 정확하고 개인화된 의료 서비스를 제공하는 방안을 탐구한다. 이 프로젝트는 제한된 의료진 자원으로 인해 어려움을 겪는 환자들에게 더 나은 의료 서비스를 제공하고자 하는 노력의 일환이다.

현재 사용 중인 의료용 AI 챗봇은 기존의 룰 기반 모델보다 더 진보된 LLM을 활용하여 개발될 것이다. 이를 통해 의료 질문에 대한 정확하고 포괄적인 답변을 제공할 수 있으며, 다양한 의료 정보 및 최신 연구 결과를 효과적으로 통합하여 제공할 수 있을 것으로 기대한다.

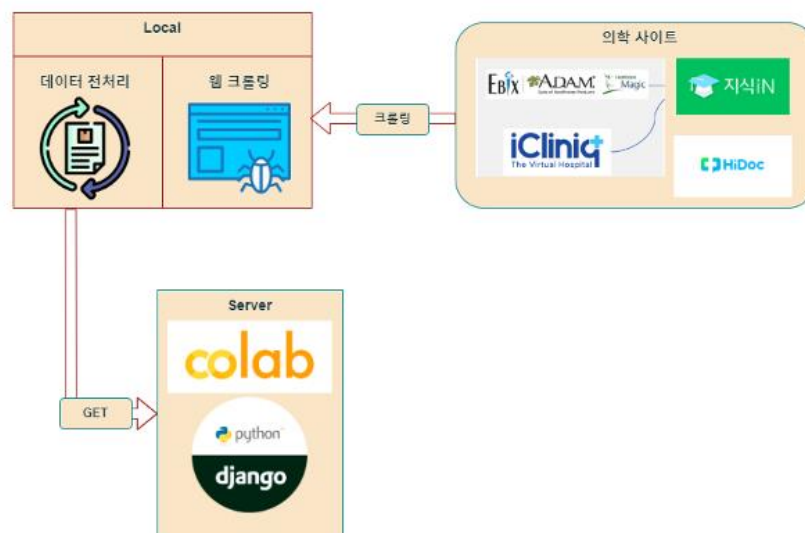
또한, 이 의료용 AI 챗봇은 환자의 개별적인 의료 상황과 요구사항을 고려할 수 있도록 개인화 기능을 강화할 것이다. 환자의 증상, 의견, 감정, 과거 의료 기록 등을 고려하여 더 정확한 의료 조언을 제공하고, 환자가 더 나은 의료 서비스를 받을 수 있도록 돕는 역할을 수행할 것이다.

이러한 개발을 통해 의료용 AI 챗봇은 의료 서비스의 접근성을 향상시키고, 의료진 자원의 한계를 어느 정도 극복할 수 있을 것으로 기대된다.

## 2. 연구 배경

### 2.1. 데이터 수집(data crawling)

여러 의학적 질문들이 게시되는 다양한 웹사이트에서 데이터 크롤링을 수행하여 JSON 파일 형식으로 정보를 저장했다. 구체적으로, 하이닥(HiDoc), 네이버 지식인, 그리고 HealthCareMagic 웹사이트에서 약 10만 건 정도의 데이터를 획득할 수 있었다. 크롤링 작업의 주요 흐름은 아래와 같다.



[그림 1] 웹 크롤링 흐름도

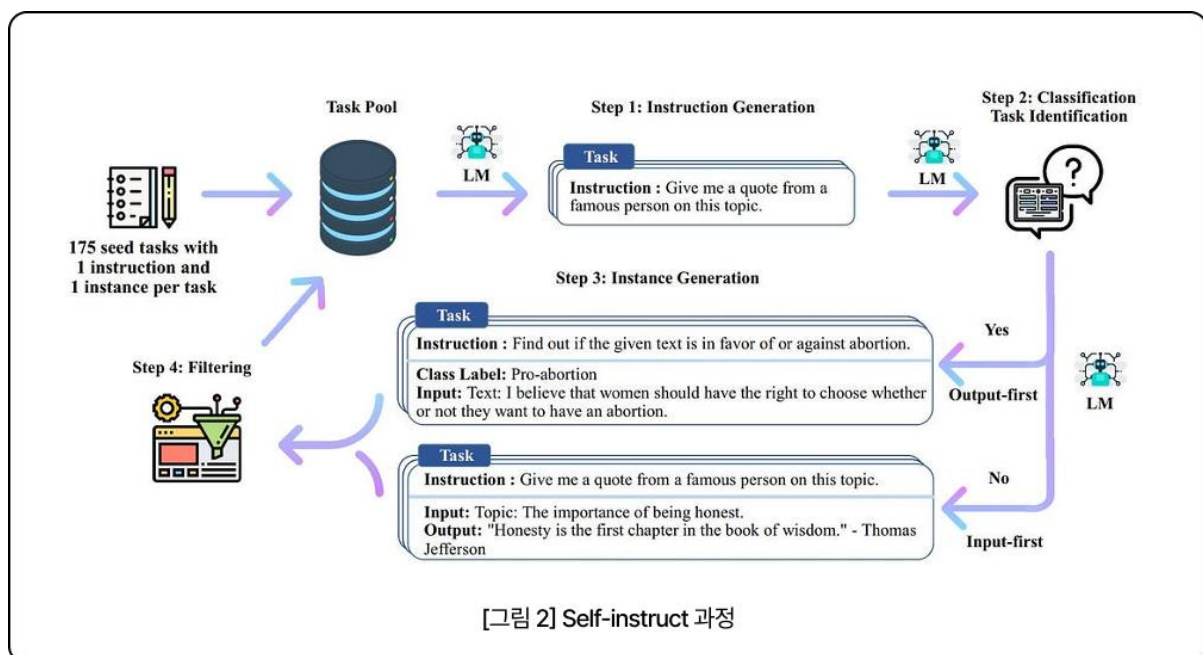
## 2.2. 배경 지식

### ○ Stanford Alpaca

Stanford Alpaca는 Stanford NLP 그룹에서 개발한 대규모 언어 모델(LLM)로, 기존의 대형 언어 모델인 ChatGPT나 Bing Chat과 같은 모델들이 겪는 문제점을 개선하고자 개발되었다. (학습에 상당한 리소스가 필요하며, 모델의 성능이 주로 인간이 만든 학습 데이터의 양과 품질에 크게 의존한다는 점)

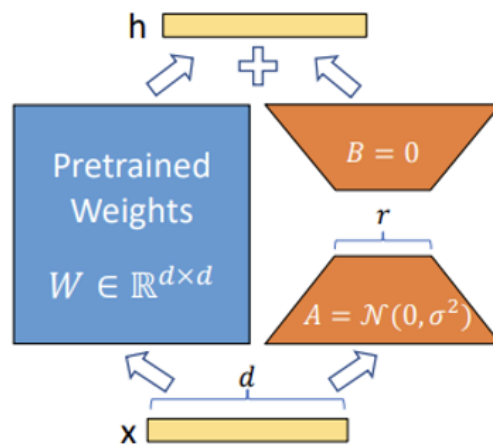
Stanford Alpaca는 이러한 문제를 해결하기 위해 설계되었으며, LLaMA는 더 작은 모델을 기반으로 하여도 유사한 성능을 얻을 수 있는 instruction tuning(지시어 조정)을 적용한 모델을 제공한다

여기서, Instruction tuning은 모델을 미세하게 조정하여 특정 작업이나 목표에 더 적합하게 만드는 프로세스를 의미한다. 이를 통해 사용자는 자신의 목적에 맞게 모델을 조정하고, 더 정확하고 효과적인 결과를 얻을 수 있다. 따라서 Stanford Alpaca는 이러한 instruction tuning을 통해 기존의 LLM 모델에 비해 효율적이며, 라이선스 문제로 인해 데이터 및 학습 모델에 접근하기 어려웠던 상황에서도 유용한 대안을 제공한다.



## ○ LoRa

LoRa는 Low-Rank Adaptation의 약자로, 적은 양의 데이터로 대규모 언어 모델(LLM)을 미세 조정하는 방법이다. 구체적으로 설명하면, LoRa는 기존 pretrained 가중치는 frozen 한 뒤 몇 개의 dense(fc) layers만 학습하는데, 이 때 dense layer의 가중치를 low rank decomposition 한 matrices 만을 최적화하는 방식이다.



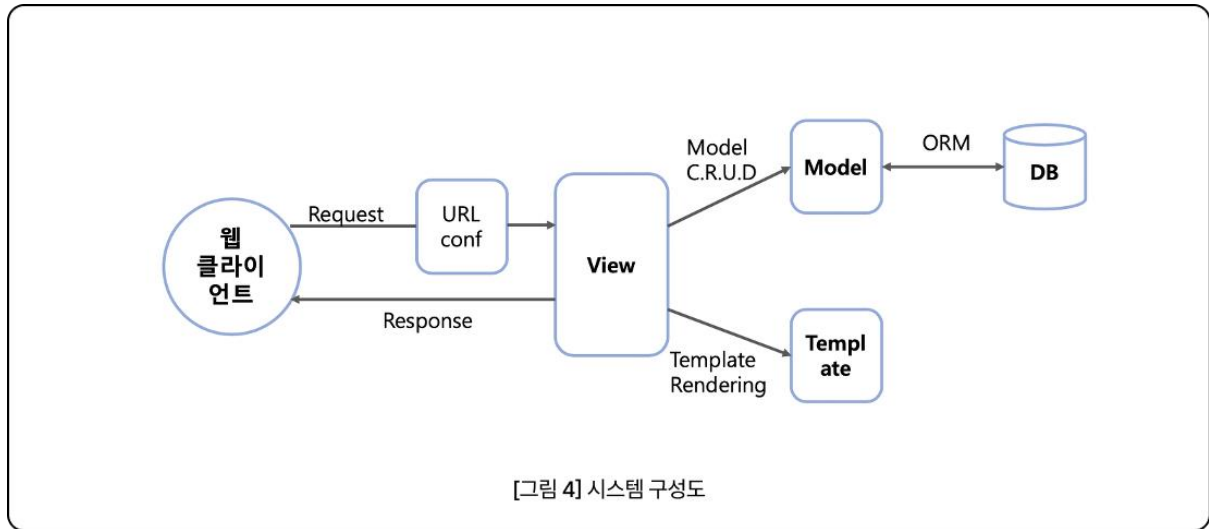
[그림 3] LoRa의 weight update 방식

위 [그림 3]에서 fine-tuning 시 pretrained 된 가중치  $W$ 는 frozen 한 뒤 low rank decomposition 된 가중치  $A$ ,  $B$ 만 학습한 후  $W$ 에 더한다. 이는 기존  $W$ 에 비해 훨씬 작은 크기의 가중치이므로, 계산 시 소요되는 리소스를 줄일 수 있다.

또한, 특정 task에 adaptation 시키기 위해  $A$ 와  $B$ 만 storage에 저장한 후 변경하면 되기 때문에, task-switching 면에서도 효율적이며, inference 시 성능이 낮아지지 않는다.



## ○ Django MVT 패턴



Django는 MTV 패턴(Model Template View)이 적용된 웹 프레임워크다.

### - Model(모델)

MVC 패턴의 모델에 대응되며 DB에 저장되는 데이터를 의미한다. 모델은 클래스로 정의되며 하나의 클래스가 하나의 DB Table이다. 장고는 ORM(Object Relational Mapping)기능을 지원하기 때문에 파이썬 코드로도 DB를 조작할 수 있다.

### - Template(템플릿)

MVC 패턴의 뷰에 대응되며 유저에게 보여지는 화면을 의미한다. 장고는 뷰에서 로직을 처리한 후 html 파일을 context와 함께 렌더링하고 이 때의 html 파일을 템플릿이라 칭한다. 장고는 자체적인 Django Template 문법을 지원하며 html 파일 내에서 context로 받은 데이터를 활용할 수 있다.

### - View(뷰)

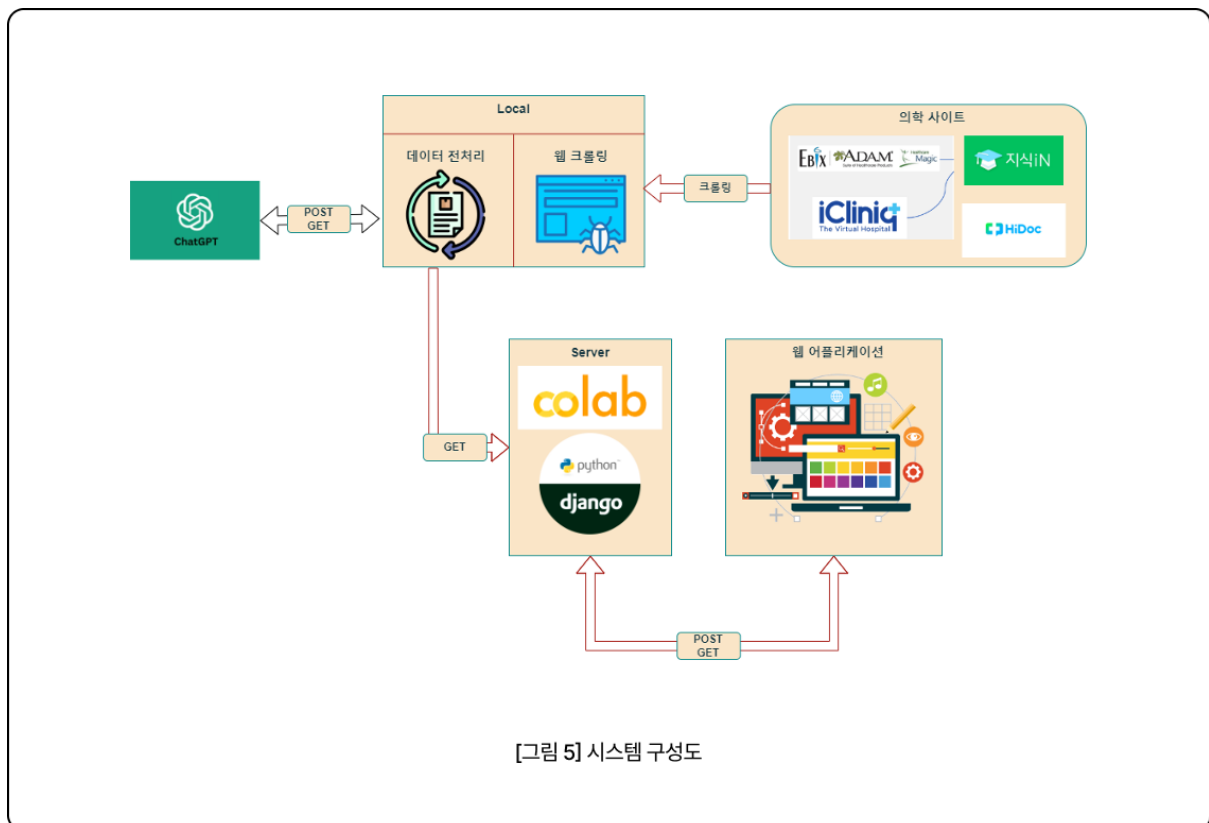
MVC 패턴의 컨트롤러에 대응되며 요청에 따라 적절한 로직을 수행하여 결과를 템플릿으로 렌더링하며 응답하거나, 백엔드와 데이터를 주고받는다.

## ○ Google Colab

구글이 제공하는 클라우드 기반 Jupyter Notebook 환경이다. 웹 브라우저에서 Python 코드를 작성하고 실행할 수 있으며, 머신러닝, 딥러닝, 데이터 분석 등의 작업을 수행할 때 매우 유용하다.

### 2.3. 시스템 구성

최종적으로 학습시킨 의료용 AI 챗봇을 웹 채팅 형태로 제공하기 위한 전체 시스템 구성도이다. 의료 데이터 확보를 위해 웹 크롤링 및 데이터 전처리를 하였고, Django 프레임워크를 이용한다. 해당 구성도는 아래와 같다.

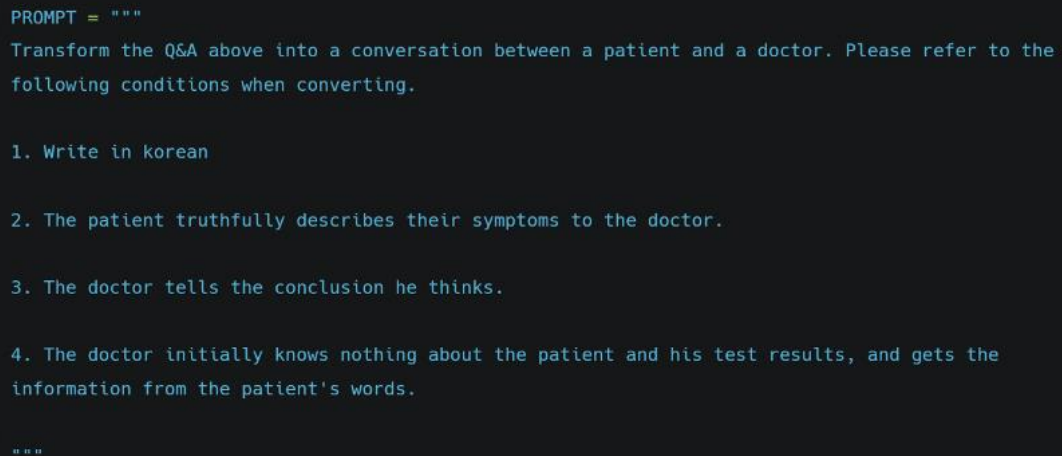


---

### 3. 연구 내용

#### 3.1. 데이터 전처리

앞서 언급한 바와 같이 크롤링한 데이터를 가공하였다. 하이닥과 같은 한국 사이트에서 확보한 데이터는 학습에 도움이 되지 않는 특수 문자나 의미 없는 문자들이 반복되는 경우를 제거하였고, 외국 사이트에서 확보한 데이터의 경우 적절한 prompt를 주고 chatgpt를 이용해 가공하였다.



```
PROMPT = """
Transform the Q&A above into a conversation between a patient and a doctor. Please refer to the
following conditions when converting.

1. Write in korean

2. The patient truthfully describes their symptoms to the doctor.

3. The doctor tells the conclusion he thinks.

4. The doctor initially knows nothing about the patient and his test results, and gets the
information from the patient's words.

"""
```

[그림 6] 데이터 가공을 위한 prompt

```

{
  "instruction": "의사의 경우 환자의 설명을 바탕으로 의학적 질문에 답변해 주세요.",
  "input": "오늘 아침에 일어났는데 앉아있을 때 방 전체가 빙글빙글 도는 느낌이 들었습니다. 불안정한 걸음  
으로 화장실에 갔는데 전동하려고 하니 메스꺼움이 느껴졌습니다. 구토를 시도했지만 토할 수가 없  
었습니다. 파나돌을 먹고 몇 시간 자고 나면 여전히 같은 느낌입니다. 그런데 누웠을거나 앉아있으  
면 머리가 안 돌아가고 움직이고 있을 때만 온 세상이 돌아가는 느낌이 듭니다.  
이전에 안경을 착용 후 회전이 들어들었으므로 연결되어 있는지  
우연의 일치인지 잘 모르겠습니다.",
  "output": "안녕하세요, 문의해 주셔서 감사합니다. 증상의 가장 유력한 원인은 말초성 현훈의 일종인 양성  
반작성 위치성 현훈(BPPV)입니다. 이 상태에서 가장 흔한 증상은 어지럼증 또는 현기증이며, 움직  
일 때 증상이 악화됩니다. 메스꺼움과 구토가 동반되는 경우도 흔합니다. 이 증상은 귀의 문제  
로 인해 발생하며 며칠 내에 저절로 개선됩니다. 베타히스틴 정제가 증상 완화에 도움이 될 수 있  
습니다. 전정 재활이나 적응 운동을 하면 이러한 증상의 재발을 예방할 수 있습니다. 이비인후과  
검진도 도움이 될 것입니다. 도움이 되길 바랍니다."
}

```

[그림 7] 가공된 데이터 예시 1

다음은 하이닥에서 수집한 데이터를 가공한 것이다.

```

{
  "instruction": "해당 설명에 알맞은 의학적 답변을 제공해주세요.",
  "input": "누런 가래가 계속 나오요..한 1년은 된거 같은데..계속 가래가 목에 낀것 같아서 가정으로 빼고있  
어요..심부전증에 흉골중수술도 했고 얼마전에는 비염수술도 했어요",
  "output": "안녕하세요, 비염수술도 얼마 전에 받았다고 하기는 했지만 증상으로 보서는  
만성 부비동염(축농증)으로 인한 후비루 이거나 만성 기관지염으로 인한 가래로 추정됩니다. 일단  
가슴 엑스레이 사진에서 문제가 없다면 PNS류를 찍어서 축농증 여부를 확인해 보고 치료를 받도록  
하세요. 경우에 따라서는 천식일 수도 있으나 위의 두 가지 경우가 아닐 경우 의심해 볼 수 있습니  
다. 감사합니다."
}

```

[그림 8] 가공된 데이터 예시 2

ChatGPT API의 활용에 제한이 있어서, 데이터 처리를 여러 번에 걸쳐 했다.

결과적으로, 약 한 달 동안 약 10만 건의 의료용 질문과 답변 쌍을 획득할 수 있었다.  
기본 모델로 Alpaca 7B 모델을 사용하며, 다른 fine-tuning 모델들을 참고하여 instruction을  
부여했다.

### 3.2. 모델 선정

처음에 사용하려고 고려한 모델은 EleutherAI의 Polyglot 12.8B 모델이었다. 고려한 이유는 이 모델이 한국어 능력이 우수하다는 평가를 받기 때문인데, Google Colab에서 이 모델을 학습시키는 데 어려움이 있었다.

모델의 크기가 상당히 크기 때문에 학습 시간이 오래 걸릴 뿐만 아니라, 효과적인 fine tuning을 위해 많은 학습 데이터가 필요했다.

위와 같은 이유로, 답변을 생성할 때 4비트 양자화를 적용해야 했으며, 이로 인해 답변의 품질이 하락하는 문제가 발생했다. 이러한 이유로 Polyglot 12.8B 모델은 fine tuning에는 적합하지 않다고 결론을 낼 수 있었다..

```
(gme) dudtkd1221@instance-1:~/my_alpaca$ nvidia-smi
Wed Aug 2 13:33:26 2023
+-----+
| NVIDIA-SMI 535.54.03                  Driver Version: 535.54.03   CUDA Version: 12.2   |
+-----+-----+-----+-----+-----+-----+
| GPU  Name      Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|=====+=====+=====+=====+=====+=====+
|   0   NVIDIA L4             On   | 00000000:00:03.0 Off |           0         |
| N/A   77C    P0              70W /  72W | 19360MiB / 23034MiB |      99%    Default |
+-----+-----+-----+-----+-----+-----+
|                                     | MIG M.         | N/A               |
+-----+-----+-----+-----+-----+-----+

[그림 9] Polyglot 12.8B 모델 vram 사용량
```

다음으로 고려한 모델은 위 12.8B 모델보다 크기가 작은 Polyglot 5.8B 모델이었다. 해당 모델의 경우 학습이나 답변 생성에 별다른 문제는 없었으나, 다른 Alpaca 모델들과 비교해 한국어 성능에서 의미 있는 차이를 보이지는 않았다. 최종적으로 이전에 학습했던 모델도 활용하기 위해 Alpaca 7B 모델을 fine tuning하기로 결정하였다.

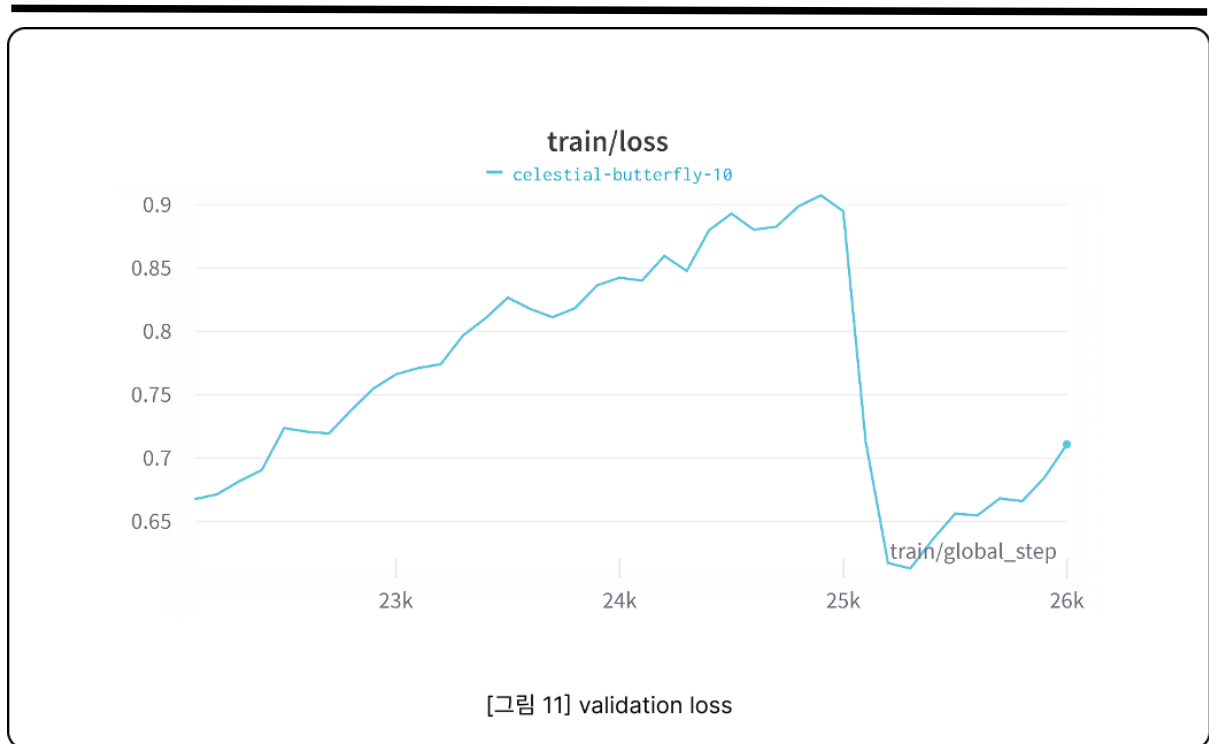
### 3.3. 모델 학습

모델 학습은 Google Colab의 A100 40G gpu를 이용하여 진행하였다. Validation 데이터셋은 전체의 5% 정도에 해당하는 5000개를 사용하였다. 약 7 epoch 이후에는 training loss는 감소하지만 validation loss는 진동하는 것을 확인하여 overfitting으로 판단하고 학습을 중단하였다. 사용한 하이퍼 파라미터와 학습 결과는 다음과 같다.

하이퍼 파라미터	설정값
Per_device_train_batch_size	4
Gradient_accumulation_steps	4
Optim	Adam
Learning_rate	2e-4
Max_grad_norm	0.3
Max_steps	26000
Warmup_ratio	0.03
Lr_scheduler_type	constant

[표 1] 모델 학습 하이퍼 파라미터





위 그래프를 통해 알 수 있듯이, training loss는 지속적으로 감소하는 반면, validation loss의 경우 학습을 계속 진행해도 0.6 ~ 0.9 사이의 값에서 진동하며 수렴하지 않는 현상을 관찰할 수 있다.

이러한 현상을 해결하기 위해 하이퍼 파라미터를 수정해 보거나, 추가적인 학습 데이터 확보 등의 방법이 있을 것으로 판단된다.

하지만 학습에 많은 시간과 컴퓨팅 자원이 들어가고, 마찬가지로 학습 데이터 확보 및 가공에도 어려움이 있어 추가적인 학습을 진행하기는 어려워 학습을 중단하였다.

---

### 3.4. 답변 생성

학습시킨 모델을 이용하여 생성 함수를 정의하고 샘플 입력을 주어 의도한대로 동작하는지 확인한다. 사용한 샘플 텍스트와 prompt, generation 함수의 하이퍼 파라미터는 다음과 같다. 실제 동작 확인에는 gradio sample code를 이용하였다.

**Sample\_text = “두통이 심할땐 어떻게 해야 하나요?”**

하이퍼 파라미터	설정값
Max_new_tokens	512
Temperature	0.5
Top_k	200
Repetition_penalty	1.1

[표 2] 생성 함수 하이퍼 파라미터

**Max\_new\_token**의 경우 최대 생성 가능한 답변 토큰 수이고, **Temperature**는 답변의 다양성을 나타낸 값이며 1에 가까울수록 모델이 다양한 답변을 한다.

**Top\_k**는 다음 토큰을 선택할 때 확률이 가장 높은 k개 중에서 선택한다는 의미이고, **Repetition\_penalty**는 같은 답변을 반복할 때 페널티를 주어 의미 없는 말이 반복되는 경우를 방지하기 위함이다.

해당 모델을 실제 활용하게 될 경우 답변이 너무 짧다고 생각되면 **Max\_new\_tokens**을 늘리고, 답변이 다양하지 못하다면 **Temperature**나 **Top\_k**를 바꿔보고, 같은 말을 여러 번 반복하는 경우 **Repetition\_penalty**를 크게 주면 된다.



```
"환자가 의사에게 아픈 곳에 대해 문의합니다.\n\n"
"환자의 문의 내용에 대해 답변하세요. 환자의 질병을 진단하고, 가능하면 처방을 하세요. \n\n"
f"### 문의:\n{instruction_input_json['instruction']}\n\n### 응답:"
```

[그림 12] generation prompt

Chatbot

두통이 심할면 어떻게 해야 하나요?



두통은 매우 흔한 질환이고 전 인구의 90% 이상이 두통을 경험할 정도로 많습니다. 그러나 뇌 질환에 의한 두통이나 전형적인 편두통은 극소수이고 대부분이 긴장성 두통입니다. 머리 전체가 무겁고 앞이마가 멍하고 관자놀이가 욱신킨거리거나 심하면 뒷머리와 목이 뻣뻣하고 저리고 어깨까지 조이고 누르는 듯한 통증이 나타나거나 심하면 귀에서는 이명이나 혹은 소리가 들릴 수 있습니다. 여자분들의 경우 빈번하게는 배란기 증상이 있습니다. 두통은 뇌질환을 의심하는 경우는 드물지만 반복되고 불편하면 신경과에 가보시면 됩니다.

[그림 13] 답변 생성 예시

질문	"두통이 심할 땐 어떻게 해야 하나요?"
답변	"두통은 매우 흔한 질환이고 전 인구의 90% 이상이 두통을 경험할 정도로 많습니다. 그러나 뇌 질환에 의한 두통이나 전형적인 편두통은 극소수이고 대부분이 긴장성 두통입니다. 머리 전체가 무겁고 앞이마가 멍하고 관자놀이가 욱신킨거리거나 심하면 뒷머리와 목이 뻣뻣하고 저리고 어깨까지 조이고 누르는 듯한 통증이 나타나거나 심하면 귀에서는 이명이나 혹은 소리가 들릴 수 있습니다. 여자분들의 경우 빈번하게는 배란기 증상이 있습니다. 두통은 뇌질환을 의심하는 경우는 드물지만 반복되고 불편하면 신경과에 가보시면 됩니다."

[표 3] 답변 생성 예시

### 3.5. 프론트 및 백엔드

LLM 관련 코드를 파이썬으로 작성하여서 연동 시 편의성을 위해 Django Framework를 이용해 개발하였다. 화면 구성 코드는 html과 css를 이용해 작성했고, 사용자 입력 받기와 답변 대기 애니메이션 등 프론트에서 필요한 기능은 js로 구현하였다. 클라우드 서버와 데이터를 주고받고, 프론트에 모델이 생성한 답변을 다시 넘겨주는 기능은 django의 views.py에 구현하였다. 동작 화면은 아래와 같다.



[그림 14] 초기 실행 화면

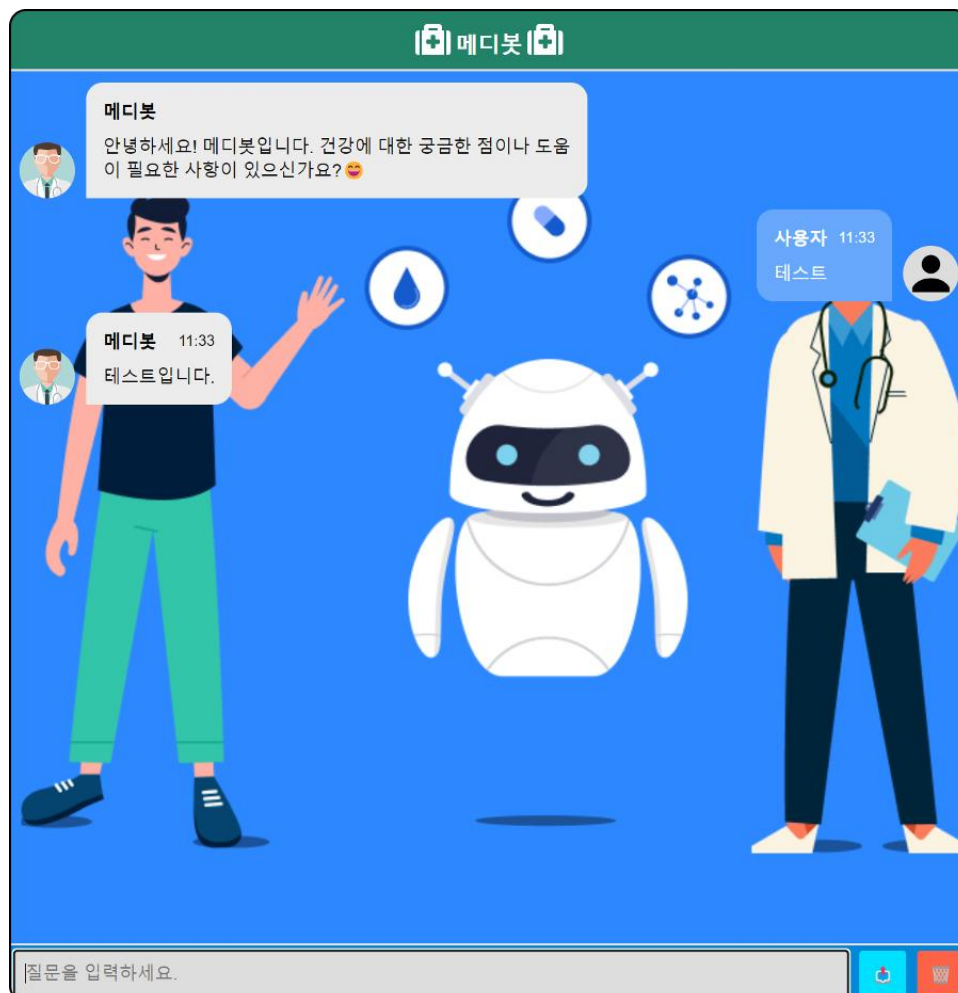


[그림 15] 사용자 입력 시 답변 대기 화면

### 3.6. API

LLM 모델의 크기가 커서 로컬 환경에서 구동 하는 것은 불가능했고, 클라우드 서버를 이용해야 했다. 원래 계획은 Amazon Web Service 나 Google Cloud Platform 같은 곳을 이용하려 했으나, Google Colab과 동일한 GPU를 가진 서버를 이용하기엔 비용의 문제가 있었다. Google Colab에는 학습을 마치고 조금 남아있는 컴퓨팅 자원도 있어서, 이것을 활용하기로 결정하였다.

Google Colab과 text를 주고받기 위해 FastAPI를 사용하였고, 생성된 데이터는 JSON 형태로 전송하였다. 또, ngrok을 이용해 외부에서 Google Colab으로 접속이 가능하게 설정하였다. 실행 결과는 아래와 같다.



[그림 16] Google Colab의 답변 출력 모습

```

class CustomFastAPI(FastAPI):
    def __init__(self):
        super().__init__()
        self.input_msg = ""
        self.add_routes()

    def add_routes(self):
        @self.post("/post_data")
        async def post_data(data: DataItem):
            print("사용자 입력:", data.message)
            self.input_msg = data.message
            response = {'status' : 'success'}
            return response

        @self.get("/get_data")
        def get_data():
            ans = "테스트입니다."
            ans_dict = {'generated_ans' : ans}
            return ans_dict

def main():
    app = CustomFastAPI()
    nest_asyncio.apply()
    public_url = ngrok.connect(8000) # FastAPI server
    print("FastAPI server is available at:", public_url)
    ▶ uvicorn.run(app, host="0.0.0.0", port=8000)

if __name__ == "__main__":
    main()

```

사용자 입력: 테스트

```

INFO:      210.101.130.131:0 - "POST /post_data HTTP/1.1" 200 OK
INFO:      210.101.130.131:0 - "GET /get_data HTTP/1.1" 200 OK

```

[그림 17] 사용자 입력을 받는 모습

## 4. 연구 결과 분석 및 평가

컴퓨팅 자원 및 시간 제한으로 인해 모델을 완벽한 최적 상태로 학습시키지는 못하였다.

현재 사용 가능한 데이터로 학습시킨 모델은 테스트 시 정확한 답변을 제공하는 경우도 있었지만, 때로는 알 수 없는 특수 문자가 생성되거나 이전에 언급한 내용을 반복하는 등 정확하지 않은 답변을 생성하기도 했다.

또한, 챗봇의 역할을 수행하기 위해 답변 속도를 단축해야 했으며, 이로 인해 최대 답변 토큰 수를 제한해야만 했다. 그러다 보니 완전한 문장이 아닌 일부분만을 생성하는 경우가 발생하였다.

그래도 진단이나 처방, 조치 등도 비교적 합리적인 선에서 답하기도 하는 등 의도한 바에 가까워지는 모습을 보여준다. 아래의 사진은 해당 정확하지 않은 예시이다.

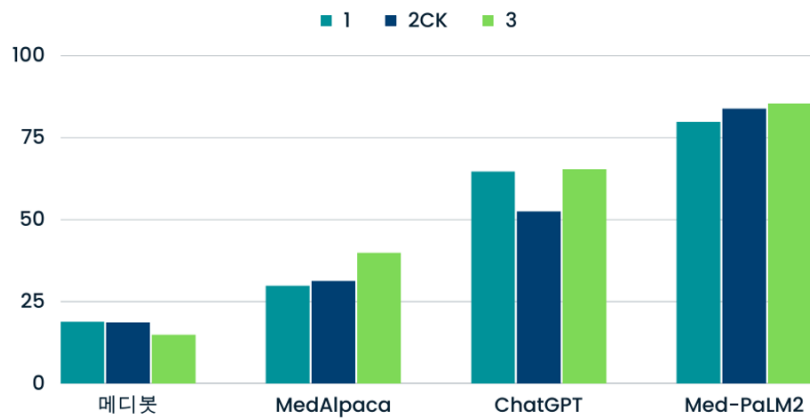


[그림 18] 답변이 손상된 모습



[그림 19] 정상적인 답변 모습

일반적으로 의료나 법률 같은 분야의 챗봇 평가 지표는 각 분야의 전문가에게 답변의 정확도 등을 문의하고 평가하는 경우가 많다. 하지만 본 과제에서는 적용하기 어려웠고, 따라서 의료용 챗봇 성능 측정 지표 중 하나인 USMLE 벤치마크를 이용하였다. USMLE 벤치마크란 모델의 미국 의사면허시험 정답률을 측정하고 순위를 매기는 것이다. 해당 벤치마크 측정 결과는 다음과 같다.



[그림 20] USMLE 벤치마크 측정 결과

같은 질문을 최대 10회까지 물어본 후 정답 여부를 측정한 결과이다. 본 과제에서 학습 시킨 모델의 경우 제대로 답변을 못하거나, 틀린 답변을 하는 경우가 많아 전반적으로 낮은 성능을 보였다. 이는 부족한 학습 데이터로 인해 모델이 충분한 의학적 지식을 갖지 못해 나타난 현상으로 보인다. 또한 대화형 모델이다 보니 context가 길어질 경우 성능이 떨어지는 문제 또한 발생한 것으로 보인다.



---

## 5. 결론 및 향후 연구 방향

본 과제에서는 LLM(Large Language Model)을 이용해 의료용 챗봇을 개발하고자 하였다. 학습을 위한 데이터는 직접 마련할 필요가 있어 하이닥과 같은 여러 의료용 질문 – 답변의 쌍을 얻을 수 있는 사이트들에서 데이터를 크롤링하고 가공하였다.

이와 같은 과정을 거치며 데이터의 수집부터 전처리, 적절한 모델 선정 및 fine tuning 과정 이후 성능 검증까지 LLM fine tuning 과정 전체를 경험해보았다. 실제로 여러 증상들을 입력해보며 답변을 확인해보았다. 다만 단순 복통도 위궤양이라고 진단하는 경우가 있는 등 정확도 측면에서 아쉬움이 있었지만, 실제 의사의 답변과 비슷하게도 답변을 하며 발전 가능성을 확인할 수 있었다. 특히, 현재 오픈소스로 공개된 모델 중 parameter 개수가 적은 편에 속하는 7B 모델을 활용한 것을 감안하면 의미 있는 성과를 얻었다고 생각한다.

비록 시간과 자원의 문제로 많은 데이터를 학습시키지는 못했지만, parameter 개수가 더 많은 모델을 충분한 데이터로 학습시킬 수 있다면 훨씬 성능이 뛰어나고 개개인에게 특화된 의료용 챗봇을 개발하고, 상용화할 수 있을 것이다.

## 6. 구성원별 역할 및 개발 일정

이름	역할
김대영	학습 데이터 수집 및 전처리 / 프론트-백 API 모델 fine tuning 및 최적화
강주호	학습 데이터 수집 및 전처리 UI/UX 디자인 설계 및 학습 결과 시각화
정영진	학습 데이터 수집 및 전처리 백엔드 구현 및 연동

[표 3] 구성원별 역할

6월				7월				8월				9월			
5	12	19	26	3	10	17	24	7	14	21	28	4	11	18	25
기본 모델 작성 및 데이터 수집/전처리															
				모델 LoRa finetuning 및 데이터 수집/전처리											
						중간 보고서									
								한국어 데이터 추가 확보 후 fine tuning, 프론트/백 구현 및 API							
													서비스 구현		
													최종 보고서 발표 준비		

[표 4] 개발 일정

---

## 7. 참고 문헌

- [1] Github. Available: <https://github.com/yizhongw/self-instruct>
- [2] Github. Available: <https://kimjy99.github.io/논문리뷰/lora/>
- [3] Velog. Available:  
  
[https://velog.io/@ssssujini99/DjangoWeb-MVC-디자인-패턴-Django 의-MTV-패턴](https://velog.io/@ssssujini99/DjangoWeb-MVC-디자인-패턴-Django-의-MTV-패턴)
- [4] Velog. Available: [https://velog.io/@yeongsang2/GCP 에서-LLM-FINE-TUNING 하기](https://velog.io/@yeongsang2/GCP-에서-LLM-FINE-TUNING-하기)
- [5] Paperswithcode. Available:  
  
<https://paperswithcode.com/paper/towards-expert-level-medical-question>
- [6] Han, T., Adams, L. C., Papaioannou, J.-M., Grundmann, P., Oberhauser, T., Löser, A., Truhn, D., & Bressemer, K. K. (Year). "MedAlpaca: An Open-Source Collection of Medical Conversational AI Models and Training Data"