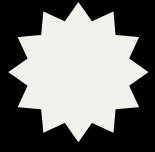




# Task 3

## 영어 상황 별 대화 ➡

Presented by  
**DSL**



# CONTENTS

■ 01 TASK 소개

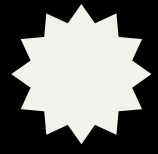
■ 04 전체 방법론

■ 02 데이터셋

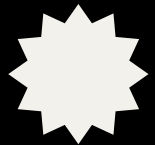
■ 05 세부 TASK별 구현 과정

■ 03 세부 TASK 설정

■ 06 최종 결과 및 Streamlit 시연



# PART1.TASK소개



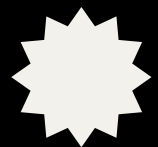
# Task 소개

## Main Task

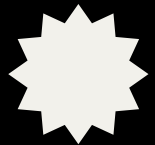
학생들의 수준에 따라 영어대화를 할 수 있는 챗봇 생성

## 조건 및 제약 사항

- 학생 수준 및 주어진 대화상황에 맞는 대화 제공
- 문법적으로 틀린 부분이 있거나 대화에서 벗어난 말을 할 경우 수정해 주도록 설계
- 일관성 있도록 프롬프트 패턴 설계



# PART2. 데이터셋



# Dataset

총 8개 학년 데이터로 구성 : 초3,초4, 중1-3,고1-3

## ■ [el3]

B: Do you like broccoli? G: No, I don't. I don't like broccoli.

## ■ [el4]

A: I want this shirt. How much is it? B: It's 6,000 won.

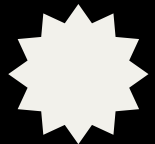
## ■ [md3]

G: My phone is not working. I need to get a new one. B: Look! Here are the new models. How about this one? G: Wow, I like it. It's very light. B: I agree. It's lighter than mine.

## ■ [md3]

A: Did you hear about the volleyball championship game tonight? B: No, I haven't. Who's playing? A: The Avengers are playing the Hurricanes. B: Really? I'm surprised that the Avengers are in the championship game. A: I know. They were the worst team last year. I believe this is going to be one interesting game.





# Dataset

## ■ [hi2]

W: We're almost done with our report on climate change. Why don't we check it together to see if everything is okay?

M: That's a good idea. Are you sure we included every graph we made?

W: Yes, they are all there in the report. They clearly show the causes of climate change that we've researched.

M: Good to hear that. How about the pictures? We've chosen the five best pictures.

W: They look very convincing. What do you think about the part discussing the students' awareness? Many of our friends participated in our survey.

M: The survey results show that many students know how serious it is. How about the action plans?

W: We have two action plans here, but I think we need more. Could you send me the data from our research?

M: Sure. I'll send the data this afternoon.

## ■ [hi3]

M: Honey, did you hear that the local aquarium is holding an event called Outstanding Octopuses?

W: Outstanding Octopuses? What's it for?

M: The purpose of the event is to promote World Octopus Day.

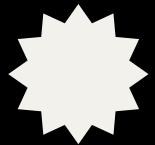
W: Sounds interesting. What programs does the event have?

M: There are several programs. They include exhibiting various octopuses from the Pacific Ocean and showing a documentary about how to protect them.

W: Really? Let me search for more information on my phone.

Oh, it's sponsored by the Aqua Life Council.

... (수능 영어 모의고사 듣기 발췌)



# Dataset

## Raw Data

대화상황 - 장래 희망에 대한 대화 예시 대화 - 다음 대화의 빈칸에 들어갈 말로 알맞은 것을 고르시오.

A: I'm interested in flying airplanes. B: Do you want to be a pilot? A: Yes, I do. How about you? B:

- 
- ①: My uncle is a pilot.
  - ②: I'm going to travel by plane.
  - ③: I'd love to, but I have no time.
  - ④: I want to be a fashion designer.
  - ⑤: I think you can become a good pilot.

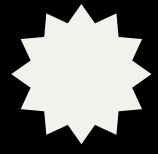
정답: ④



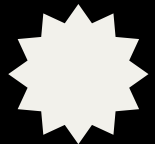
## Preprocessed Data

A: I'm interested in flying airplanes.  
B: Do you want to be a pilot?  
A: Yes, I do. How about you?  
B: I want to be a fashion designer.





# PART3. 세부 TASK 설정



# 세부 Task 설정

## 01 대화 시작/ 종료 패턴 및 내용 구성



학년별 대화 주제 자동  
설정 및 시작, 종료 패턴  
구성

## 02 상황-역할 설정



상황은 챗봇이  
system prompt로 제시 후  
대화 시작

## 03 학년별 수준 구체화

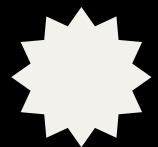


기업 제공 dataset :  
el3, el4, md1-3, hi1  
교육청 모의고사  
dataset:  
hi2-3

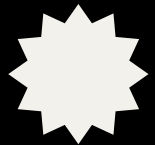
## 04 유저 발화 오류 침삭 및 피드백 제공



- ①문법 오류 표현
- ②문법은 맞지만,  
부자연스러운 표현
- ③맥락에 맞지 않는 발화



# PART4. 전체 방법론



# 전체 방법론 (1)

## 코드 구성

유저 정보:  
유저가 학년을 입력

초기 메시지:  
코드 실행 시  
GPT에 입력되는 프롬프트  
유저가 입력한 학년이 포함됨.

대화 실행 loop:  
유저의 입력을 input으로 받아  
대화 진행

01

초기 메시지에 데이터셋  
담고, 챗봇 기능을  
함수로 정의하여  
유저 인풋마다 함수 실행

함수 내 정의한 발화 마다  
API를 호출해야 하여  
대화 로딩 시간이 오래 걸림.

02

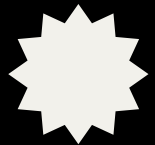
초기메시지에  
데이터셋을 담고,  
기능은 유저의 인풋 뒤에  
텍스트로 붙여 전달

문법 수정에만 너무  
집중되는 등 원활히  
대화가 안됨.

03

초기 메시지에 데이터셋  
및 기능들을 담아 한번에  
전달

세 방법 중 가장 기능 구현  
정도가 높고, 대화 구성 및  
흐름이 가장 원활함



# 전체 방법론 (2)



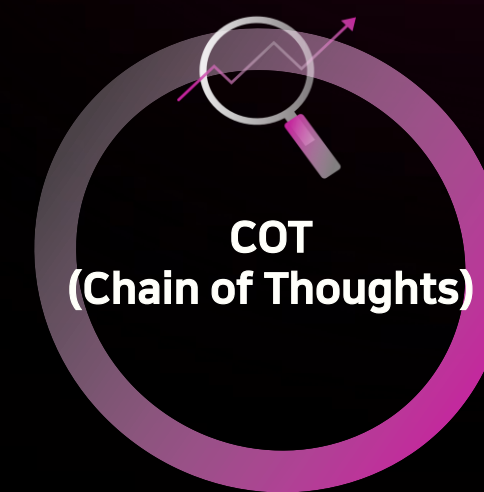
모델에게 특정 역할을 부여하여  
응답의 배경과 스타일을 제어  
"너는 학생의 영어 회화를 돕는  
영어 회화용 선생님이야."



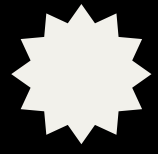
예제를 넣어 이와 비슷한 형식,  
패턴으로 답하도록 지시  
(대화상황, 학년, 대화 컬럼이  
담긴 데이터셋을 텍스트로 입력)



응답 길이 및 문체를 조정하고  
특정 정보를 포함하도록 제어  
(1~2문장으로 조절, 유저가  
문법을 틀릴 시 관련 문법 내용을  
포함하여 설명)

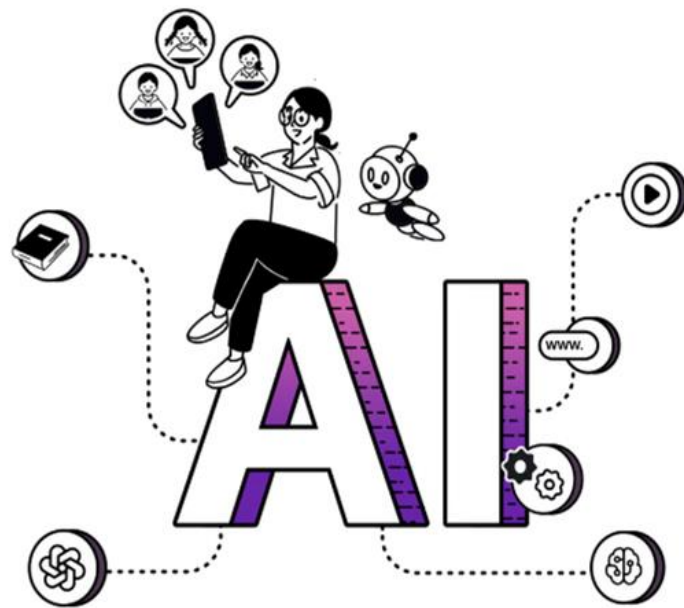
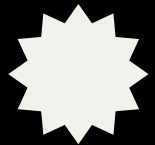


복잡한 문제를 해결할 때 논리적인  
사고 과정을 차례로 나열하여 최종  
답변을 생성하도록 모델을 유도  
ex)  
"1단계: 사용자가 'Hi, LittleLaiv!'라고  
인사하면, 먼저 대화 상황을 제시해야 해."  
"2단계: 상황을 제시한 뒤, 상황에 적합한  
영어 문장으로 대화를 시작해줘."



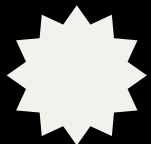
# PART5. 세부 TASK별 구현 과정





세부 Task 별 구현 과정

대화 시작/종료 패턴 및  
내용 구성



## 세부 Task 별 구현 과정

# 대화 시작/종료 패턴 및 내용 구성

- 시작, 종료 조건 부여하여 사용자가 간단하게 대화 시작과 종료할 수 있도록 설계

### 1) 대화 시작/종료 조건 부여

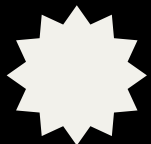
"사용자가 'Hi, LittleLaiv!'라고 인사하면 대화를 시작해."

"대화가 끝나면 '우린 {새로운 주제}를 할 거야. 대화 시작할게!'라고 말하고, 새로운 주제를 만들어 대화를 이어가."

.....

"사용자가 'Bye-bye, LittleLaiv!'라고 인사하면, 'Bye, see you later!'라고 답한 뒤 대화를 끝내줘."

"대화가 끝나면 틀린 문장이나 부자연스러운 표현을 간단히 정리해줘."



## 세부 Task 별 구현 과정

# 대화 시작/종료 패턴 및 내용 구성

기존에는 while 문안에 user\_input이 먼저 입력되어 사용자가 먼저 대화를 시작해야 했으나, response 를 while 문 밖으로 빼내어 GPT가 먼저 주제를 제시하여 말을 걸 수 있도록 프롬프팅

## 2) 대화 시작 방식 설계

```
response = openai.ChatCompletion.create(  
    model="gpt-4o",  
    messages=messages,  
    max_tokens=200,  
    temperature=1.2,  
)
```

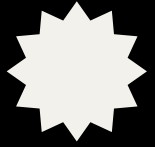
```
assistant_message = response.choices[0].message["content"]  
print("Assistant:", assistant_message)  
messages.append({"role": "assistant", "content": assistant_message})
```

```
# 대화 루프  
while True:  
    # 사용자 입력 받기  
    user_input = input("User: ")  
    messages.append({"role": "user", "content": user_input})  
  
    response = openai.ChatCompletion.create(  
        model="gpt-4o",  
        messages=messages,  
        max_tokens=200,  
        temperature=0.7,  
    )  
  
    assistant_message = response.choices[0].message["content"]  
    print("Assistant:", assistant_message)  
    messages.append({"role": "assistant", "content": assistant_message})
```

Assistant: 주말 계획에 대한 대화

안녕! 우리는 주말 계획에 대해 이야기할 거야. 대화 시작할게! What are your plans for the weekend?

User:



## 세부 Task 별 구현 과정

# 대화 시작/종료 패턴 및 내용 구성

Gpt가 랜덤하게 먼저 제시하는 주제를 더 다양하게 구체화하기 위하여 추가 프롬프팅을 진행

### 3) 대화 내용의 다양성을 위한 추가 프롬프팅

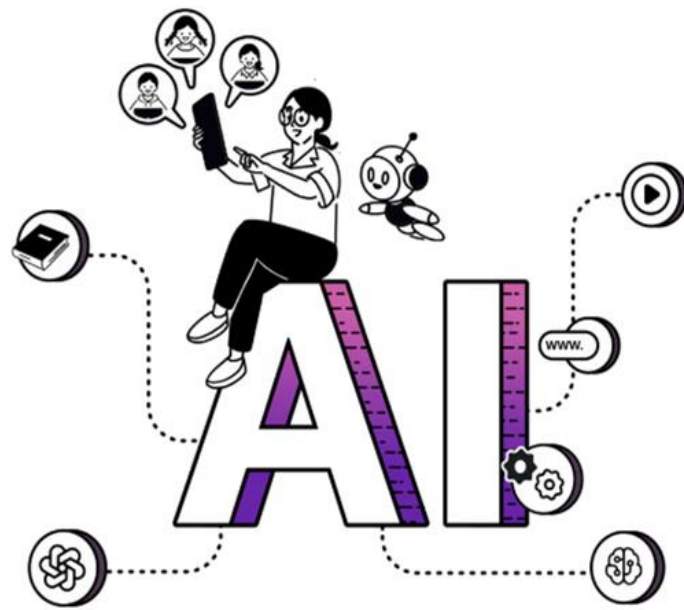
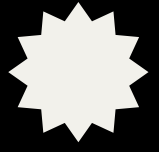
#### i) Temperature 조절

대화 주제를 제시하는 부분은 주제의 다양성을 위해 Temperature를 1.2정도로 조절

대화를 주고 받는 부분(while 문 내부)은 교육적 목적을 위해 0.7로 설정

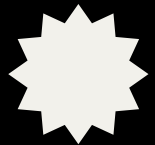
#### ii) 프롬프트 추가

초기 메시지 마지막에 "중학생이 흥미로워 할 대화주제를 제시해줘. 그럼 시작하자!" 의 한 문장을 추가했을 뿐인데, 추가하기 전보다 학생 수준에 적합한 주제 제시하는 것으로 확인



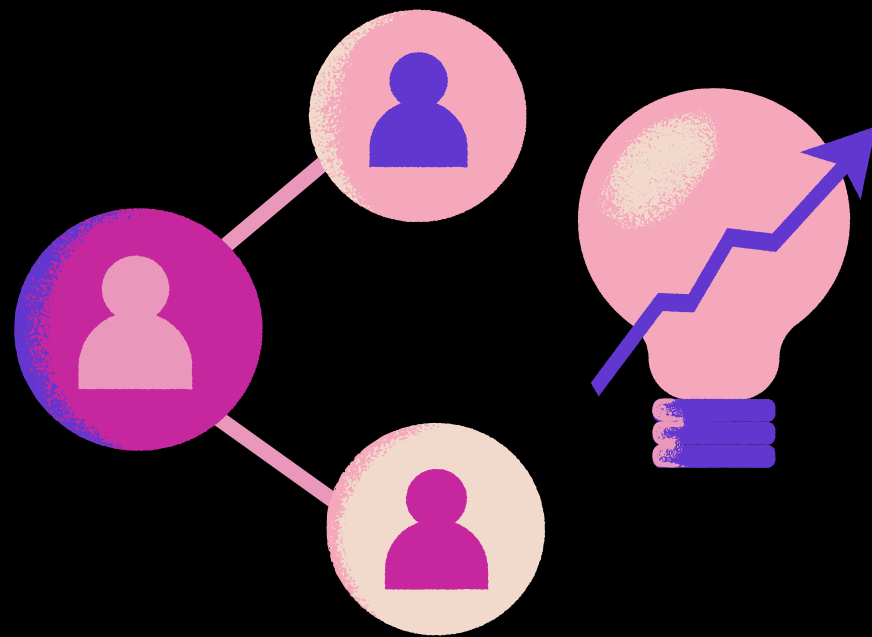
세부 Task 별 구현 과정

**상황-역할 설정**



# 세부 Task 별 구현 과정

## 상황-역할 설정



- 각 학년 수준에 맞는 역할 부여
- 형식이 부여된 주제 제시 멘트를 설정해 주는 방식으로 프롬프팅
- 주제 제시 후 먼저 대화를 시작하도록 설계
- 해당 기능에 대한 예시 제시로 일관성과 정확도 확보

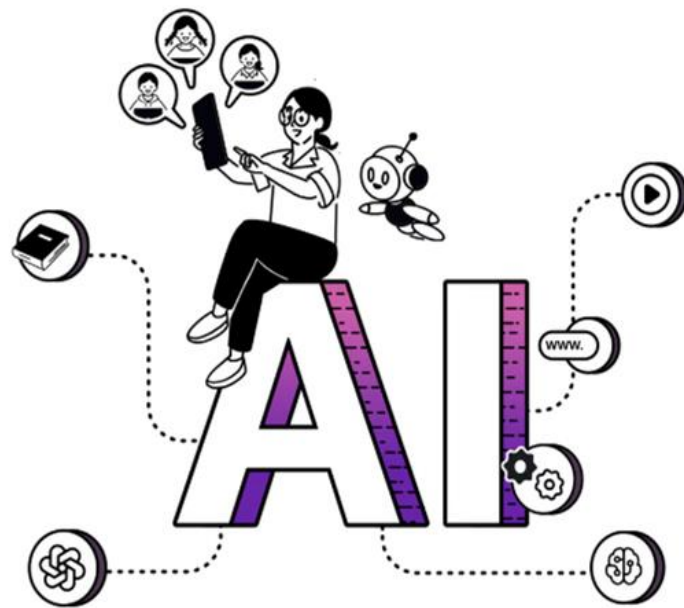
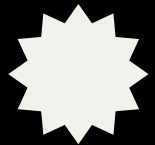
(초등학생) "너는 한국의 어린 아이의 영어 회화를 돕는 영어 회화용 선생님이야." "아이들에게 친절하게 영어를 알려주고 대화를 도와줘야 해."

"대화를 시작하기 전에, 한국어로 간단히 대화 상황을 알려줘. 대화 상황은 '{장소}'에서의 대화' 또는 '{대상}'에 대한 대화'로 말해줘."

"상황을 알려준 다음에는, 간단한 영어 문장으로 먼저 대화를 시작해줘."

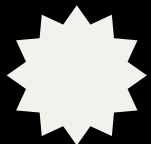
"예를 들어, '학교 축제에 대한 대화'라는 주제를 제시한 경우, '안녕! 우리 학교 축제에 대해 대화를 할 거야. 대화 시작할게! What are you most looking forward to at this festival?' 라고 대화를 시작해."





세부 Task 별 구현 과정

**학년별 수준 구체화**



# 세부 Task 별 구현 과정

## 학년별 수준 구체화

### 1) 학년별 수준 구체화

전체 데이터셋 대신, 학년별로 5개 이하로 데이터를 추출해 다음과 같은 요약 데이터셋을 만들어 프롬프트로 전달

난이도 조절 기능이 떨어지는 것으로 보아 데이터셋 부족하다고 판단.  
→ 학년별 데이터를 10개 내외로 늘림

=== 학년: el3 ===

대화 상황: 안부에 대한 대화

대화 예시: G: How are you??B: I'm good.

대화 상황: 사물에 대한 대화

대화 예시: G: What's this? B: It's a book.

...중략...

=== 학년: el4 ===

대화 상황: 요일과 일정에 대한 대화

대화 예시: A: What day is it today? B: It's Friday.  
A: Good! I have a reading class.

...중략...



중학교 1학년

Assistant: That sounds exciting! Why do you want to be a pilot?

User: It is fun

Assistant: I see, it must be fun! Have you ever flown in a plane before?

Corrected: It is fun.

설명: 문법적으로는 큰 문제는 없지만, "It is fun"을 "It must be fun"으로 바꾸면 좀 더 자연스럽게 구체적인 이유를 묻는 것과 잘 어울려. "must"는 추측이나 확신을 나타낼 때 사용해.

고등학교 1학년

User: I like sushi

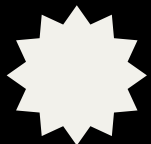
Assistant: 맥락에 안 맞아요.

예시: "I'm considering being a chef because I like sushi."

이렇게 대화 주제인 장래 희망에 맞춰 대답할 수 있습니다.

User: mabye a chef

Assistant: That's a great choice! Being a chef would allow you to explore different cuisines and create delicious dishes. Do you have a favorite dish you would like to learn how to cook?



## 세부 Task 별 구현 과정

# 학년별 수준 구체화

## 2) 프롬프트 메시지 분리를 통한 수준 구체화

GPT의 발화가 학생 수준에 적합하도록 조정해야 할 필요

→ 초등학생 저학년(el3-4), 중학생 (md1-hi1), 고등학생 (hi2-3) 초기 프롬프트를 분리하여 수준 구체화

"너는 한국의 **어린 아이의** 영어 회화를 돕는 영어 회화용 선생님이야."

"아이들에게 친절하게 영어를 알려주고 대화를 도와줘야 해."

...중략...

"사용자는 영어를 잘 못하는 **어린 아이니까**, **el3의 예시**를 참고해서 매우 쉬운 단어와 짧은 문장을 사용해. 최대 2문장으로 대화해줘."

"**어린 아이이기 때문에, 문법적 원리에 대한 설명보다는 올바른 표현이 무엇인지 위주로 설명해줘.**" "**아이들이 이해하기 쉽게 설명하고 대화를 재미있게 만들어줘. 그럼 시작하자!**"

el3, el4



"너는 한국의 **중학생의** 영어 회화를 돕는 영어 회화용 선생님이야."

"학생들에게 친절하게 영어를 알려주고 대화를 도와줘야 해."

...중략...

"사용자는 영어가 능숙하지 않은 한국의 **중학생이니까**, **md1의 예시**를 참고해서 적절한 문장과 단어를 사용해. 최대 2문장으로 대화해줘."

"**중학생이 흥미로워할 대화 주제를 제시해줘. 그럼 시작하자!**"

md1, md2

"너는 한국의 **고등학생의** 영어 공부를 돕는 영어 회화용 선생님이야."

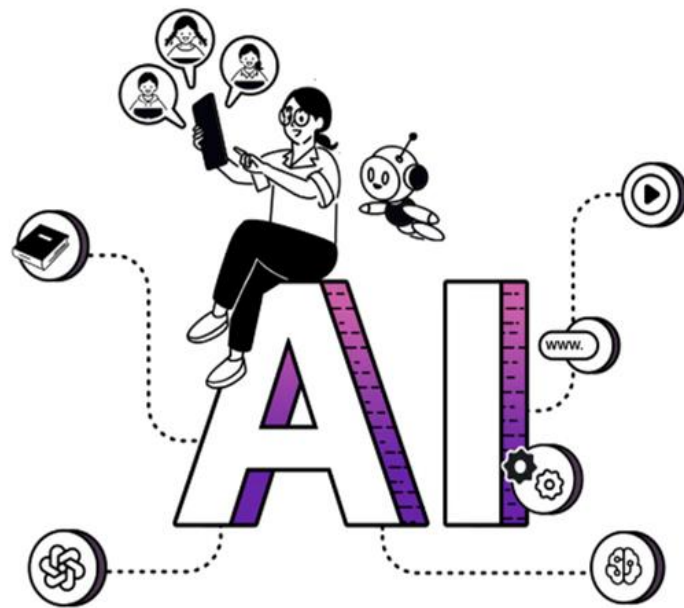
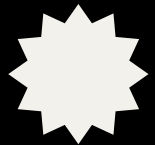
"학생들에게 친절하게 영어를 알려주고 대화를 도와줘야 해."

...중략...

"사용자는 영어가 능숙하지 않은 한국의 **고등학생이니까**, **hi2의 예시**를 참고해서 적절한 문장과 단어를 사용해. 최대 3문장으로 대화해줘."

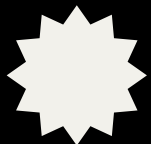
"**학생들이 이해하기 쉽게 설명하고 대화를 이어가줘. 그럼 시작하자!**"

hi2, hi3



세부 Task 별 구현 과정

**유저 발화 오류 침삭  
및 피드백 제공**



## 세부 Task 별 구현 과정

# 유저 발화 오류 침삭 및 피드백 제공

유저가 대화 맥락에서 벗어난 말을 한 경우 이를 지적해주고 예시 답안을 추천해주는 기능

유저가 학년, 대화상황을 입력 → 초기 설정 메시지에 적용되어 gpt-4o에 전체 데이터셋과 함께 전달 → 대화 진행

### 1) 맥락 피드백 기능만 있는 경우

대화 상황: 장래 희망에 대한 대화 학년: md1 대화 예시: A: I'm interested in flying airplanes. B: Do you want to be a pilot? A: Yes, I do. How about you? B: I want to be a fashion designer 대화 상황: 책에 대한 대화 학년: md1 대화 예시: A: What do you think about this book? B: A History of Black Holes? I think it'll be difficult. ...

#### 초기 설정 메시지:

너는 이제 영어 회화용 챗봇이야.

내가 상황에 안 맞는 말을 하면 맥락에 안 맞다고 말한 뒤 추천

답안 예시를 영어로 간략히 써줘.

그리고 너 또한 이전의 대화를 기억하면서 맥락에 맞는 대화를 해. 모든 말은 너무 길지 않게 간략하게 해줘. 이제 내 정보를 입력할게.

우리는 **\*\*{topic}\*\***을 하는 상황이고 내 학년은

**\*\*{student\_level}\*\***이야.

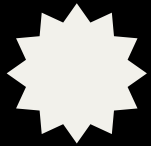
아래는 대화 상황, 학년, 대화 예시에 대한 데이터셋이고 이와 비슷한 길이와 난이도로 답변해줘.

특히 내가 입력한 대화 상황 및 학년에 적합한 수준으로 말해주면 돼.

예를 들어 md1은 중1, hi1은 고1을 뜻해.

**{dataset\_summary}**





## 세부 Task 별 구현 과정

# 유저 발화 오류 침삭 및 피드백 제공

## 결과 예시

### 장래희망에 대한 대화

중1

Assistant: That's awesome! Why do you want to be a pilot?

User: dog is cute

Assistant: 맥락에 안 맞는 것 같아요. 다시

말씀해주시겠어요? 예를 들어, "I love airplanes and traveling."

User: I love it

Assistant: That's great! Have you ever been on a plane before?

User: many times how bout u

Assistant: Yes, I've been on a plane a few times.

Do you have a favorite place you've flown to?

User: 야아아앙

Assistant: 맥락에 안 맞는 것 같아요. 다시

말씀해주시겠어요? 예를 들어, "I really enjoyed flying to Paris."

### 장래희망에 대한 대화

고1

Assistant: That's cool! Do you want to fly commercial planes or something different?

User: dog is cute

Assistant: That doesn't quite fit our conversation about future aspirations.

Try this: "I want to be a pilot because flying is fascinating."

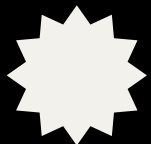
User: 야

Assistant: It seems like you're using a casual expression in Korean. Let's keep the conversation in English.

So, what interests you most about becoming a pilot?

문제점: 맥락 수정 기능 잘 수행하며 적절한 난이도로 대화하나, 문법 수정 기능 등 기타 기능 추가 필요





## 세부 Task 별 구현 과정

# 유저 문법 수정 기능 제공

유저가 문법을 틀리면 수정해주는 기능을 위해, 초기 메시지에 프롬프트를 추가

## 2) 문법 수정 기능 구현

초기 설정 메시지:

너는 이제 영어 회화용 챗봇이야.

내가 맥락은 맞는데 문법을 틀리면, 우선 대화를 이어나가준 뒤 그 밑에 고친 문장을 간략히 써주고, 한글로 해당 문법에 대한 내용과 함께 틀린 이유를 간략히 설명해줘.

만약 내가 대화 주제 및 상황에 많이 안 맞는 말을 하면, 맥락에 안 맞다고 한국어로 말한 뒤, 추천 답안 예시를 영어로 간략히 써줘. 이 모든 설명은 유저의 학년을 고려해서 적합한 수준으로 설명해주어야 해.

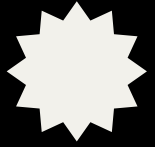
내 정보를 입력할게.

우리는 **\*\*{topic}\*\***을 하는 상황이고 내 학년은 **\*\*{student\_level}\*\***이야.

아래는 대화 상황, 학년, 대화 예시에 대한 데이터셋이고, 이와 비슷한 길이와 난이도로 답변해줘. 특히 내가 입력한 대화 상황 및 학년에 적합한 수준으로 말해주면 돼.

예를 들어 md1은 중1, hi1은 고1을 뜻해.

{dataset\_summary}



## 세부 Task 별 구현 과정

# 유저 문법 수정 기능 제공

## 결과 예시

### 장래희망에 대한 대화

#### 중1

Assistant: Sure! What do you want to be in the future?

User: I want pilot

Assistant: That's an exciting dream! Do you want to fly planes all around the world?

Corrected: I want to be a pilot.

설명: "want" 다음에는 동사 원형이 오며, "I want to be a pilot."와 같이 "to be"를 사용해야 합니다. "I want pilot."는 "pilot"이 명사이므로 적절하지 않습니다.

### 장래희망에 대한 대화

#### 고1

User: hi

Assistant: Hello! How can I assist you today? Are we talking about your future hopes and dreams?

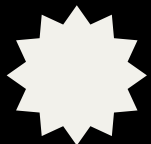
User: I want pilot

Assistant: That's great! Do you want to be a pilot in the future?

Corrected sentence: "I want to be a pilot."

설명: 영어로 직업이나 희망을 말할 때 "to be"를 사용해야 해. "I want" 다음에는 동사의 원형이 와야 하므로 "to be a pilot"가 맞아.

문법, 맥락 수정, 난이도 조절 기능 잘 구현됨. 중1과 달리 고1에선 'aviation' 등 보다 어려운 단어 사용.  
BUT 문법 수정 시 대화가 중간에 중단되어야 한다는 문제 있음



## 세부 Task 별 구현 과정

# 유저 문법 수정 기능 제공 - 개선 과정

조건 분기 명시를 통해 문장이 문법적으로 틀렸는지, 아니면 부자연스러운 표현인지를 구분하여 대응 방식을 제시

대화 종료 후 전체 수정 사항에 대한 대화 피드백을 추가 제시

결과 형식을 명시해 출력 구조를 명확히 지정해 출력 일관성을 높임

## 2) 문법 수정 기능 구현

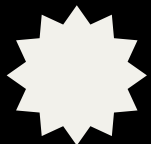
"내가 문법적으로 틀린 말을 하면, 맞는 표현으로 고쳐서 다시 물어봐주고, 고친 부분은 큰 따옴표로 표시해줘. 예를 들어 내가 I want be a doctor라고 하면 Oh, you want to be a doctor와 같이 먼저 고쳐서 말해준 다음 대화를 이어가줘." / "내가 문법적으로는 틀리지 않지만 원어민들이 사용하지 않는 부자연스러운 표현을 사용하면, 자연스러운 표현으로 고쳐서 그걸 의미한 건지 다시 물어봐 주고, 고친 부분은 큰 따옴표로 표시해줘." / "묻는 내용에 어울리지 않는 대답을 하거나 대화 맥락에 맞지 않는 듯한 말을 하면, 어떤 걸 말하려고 한건지 물어서 나에게 확인한 다음에 다시 대화를 이어가 줘. 이 때 내가 말하고자 한 걸로 추측되는 내용이 있다면 적절한 표현으로 고쳐서 그 내용을 말하려고 한 게 맞는 지 확인해줘"

.....

"대화를 마무리한 후, 지금껏 나눈 대화 중 수정해준 내용들은

1. 내 틀린 표현, 2. 고쳐진 알맞은 표현, 3. 문법 원리 및 고쳐야 하는 이유 4. 해당 표현을 연습할 수 있는 연습 문제의 구성으로 피드백을 제시해줘."

"이 모든 설명은 한글로 써주고, 유저의 학년을 고려해서 적합한 수준으로 설명해주어야 해."



## 세부 Task 별 구현 과정

### 유저 문법 수정 기능 제공 - 개선 과정

#### 결과 예시

User: Teacher is a great job, but also a very tiring job.

Assistant: Oh, you think being a teacher is a great job, but also a very tiring one? What makes you feel that way?

User: Students' parents draw teacher.

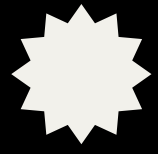
Assistant: I see what you're trying to say. Do you mean "Students' parents put pressure on teachers"?

지금까지 대화 중에서 고쳐진 표현들을 정리해 줄게:

1. 틀린 표현: Students' parents draw teacher.
2. 고쳐진 표현: Students' parents put pressure on teachers.
3. 문법 원리 및 고쳐야 하는 이유: "draw"는 '그리다'라는 의미로 문맥에 맞지 않으며, "put pressure on"은 '압박을 가하다'라는 의미로 이 상황에 적합합니다.
4. 해당 표현을 연습할 수 있는 연습 문제: 다음 문장을 완성해 보세요: "The coach often \_\_\_\_\_ pressure on the players to perform well." (힌트: '압박을 가하다'라는 의미의 표현 사용)

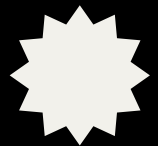
필요할 때 언제든지 도움을 요청해!

대화 흐름을 끊지 않고 자연스럽게 잘못된 표현에 대한 수정이 가능하도록 수정됨  
또한, 문법적으로 맞더라도, 전체 대화 맥락을 고려하여 맞지 않는 표현을 잡아낼 수 있게 개선됨  
피드백의 형식이나 구체적인 제시 워딩은 기업이 원하는 형태로 변형하여 제시 가능



# PART6. 최종 결과 및 Streamlit 시연






# Streamlit 시연 - 초등학생용 챗봇

Settings

Enter your OpenAI API key:

.....



**LA.IV DATA**  
라이브데이터

## English Conversation Chatbot for Elementary School Students

'Hi, LittleLaiv!'를 입력하여 대화를 시작해주세요.  
'Bye-bye, LittleLaiv!'를 입력하여 대화를 마무리해주세요.

### Chat

---

### Your Input

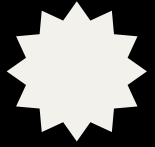
Type your message here:

Hi, LittleLaiv! |

Press Enter to apply

Send





# Streamlit 시연 - 초등학생용 챗봇

### Settings

Enter your OpenAI API key:

.....

### Chat

You: Hi, LittleLaiv!

Bot: Hello! How are you today?

You: I'm god

Bot: 아, 'I'm god'는 영어에서 잘못된 표현이에요. '고맙다, 고마워'는 'Thank you, Thanks' 라고 말해요. 'I'm good' 이라는 영어 표현은 '나는 잘지냈다' 라는 뜻이에요. 영어로 맞게 고쳐보면 'I'm good'가 될 거예요. 이제 올바른 표현으로 다시 한 번 말해볼까요?

I think you meant to say, "I'm good." Try again!

You: I'm good.

Bot: Great! What's your favorite color?

You: pink

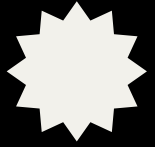
Bot: Oh, pink! That's a lovely color. Do you have a pink shirt?

You: 맞아 있어

Bot: Great! In English, you can say, "Yes, I do." Shall we talk about animals next? Let's start! Do you like dogs?

---

Your Input



# Streamlit 시연 - 중학생용 챗봇

Enter your OpenAI API key:

.....

'Hi, LittleLaiv!'를 입력하여 대화를 시작해주세요.  
'Bye-bye, LittleLaiv!'를 입력하여 대화를 마무리 해주세요.

## Chat

You: Hi, LittleLaiv!

Bot: 안녕! 오늘은 '도서관에서의 대화'를 해보려고 해. 대화를 시작할게! Have you finished reading the book you borrowed last week?

You: Yes, I do.

Bot: 그 문장은 조금 부자연스러워. 'Yes, I do.'는 '나는 그렇게 한다'는 뜻이에요. 'Yes, I have.'가 '그래, 내가 그 책을 다 읽었어.'라는 뜻으로 더 적합해요. Let's try again: Have you finished reading the book you borrowed last week?

You: Yes, I have.

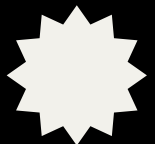
Bot: Great! Did you enjoy the book? What part of the book did you like the most?

You: 난 마지막 챕터가 가장 재밌어

Bot: That's great! But let's try to answer in English. You can say: "I enjoyed the last chapter the most." Can you try that?

You: I enjoyed the last chapter the most.

Bot: Perfect! That book has an interesting plot twist at the end, doesn't it?



## Streamlit 시연 - 중학생용 챗봇

Bot: Perfect! That book has an interesting plot twist at the end, doesn't it?

You: Bye-bye, LittleLaiv!

Bot: Bye, see you later!

오늘 대화하면서

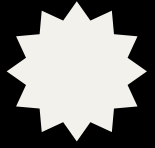
1. 문법적으로 틀린 경우:

- 사용자가 작성한 원래 문장: "Yes, I do."
- 고쳐진 문장: "Yes, I have."
- 간단한 이유: 영어에서 현재완료형을 사용할 때 'have'를 사용해야 해요.

2. 부자연스러운 표현 사용:

- 사용자가 작성한 원래 문장: "난 마지막 챕터가 가장 재밌어"
- 고쳐진 문장: "I enjoyed the last chapter the most."
- 간단한 이유: 영어로 대화를 하는 동안에는 한국어를 사용하면 안 돼요.

잘 따라와 주셔서 감사해요. 다음 회차에 또 만나요!



# Streamlit 시연 - 고등학생용 챗봇

### Settings

Enter your OpenAI API key:

### Chat

**You:** Hi, LittleLaiv!

**Bot:** 안녕! 오늘 우리는 '도서관에서의 대화'를 할 거야. 시작할게! "Hi there, are you looking for a specific book?"

**You:** Yes

**Bot:** Great! Can you tell me the title of the book?

**You:** 채식주의자

**Bot:** In English, we say, "The Vegetarian". So you can say, "Yes, I'm looking for 'The Vegetarian'". Let's try it.

**You:** Yes, I'm looking for 'The Vegetarian'.

**Bot:** Perfect! "Okay, do you know the author's name?"

**You:** No, I doesnt

**Bot:** 언어학습을 할 때 일반적으로 "doesn't" 는 3인칭 단수 주어와 함께 쓰입니다. 예시로는 "he", "she", "it" 있습니다. 또한 문장의 끝에는 구두점을 잊지말고 써주세요. 그래서 "No, I don't." 라고 해야 돼요. 다시 시도해볼까요?

**You:** No, I don't.

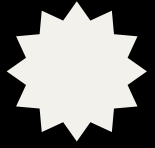
**Bot:** Great job! "That's no problem. Let me help you find it. Follow me." 이제 우리는 '마트에서의 대화'를 할 거야. 대화 시작할게! "Excuse me, where can I find the eggs?"

**You:** 2층 오른쪽 코너에 있어요

**Bot:** That's correct in Korean, but let's try it in English. You can say, "They are at the right corner on the second floor.". Let's try it again!

**You:** They are at the right corner on the second floor.





## Streamlit 시연 - 고등학생용 챗봇

### Settings

Enter your OpenAI API key:

**You:** They are at the right corner on the second floor.

**Bot:** Fantastic! "Thank you! And could you also tell me where the milk is?"

**You:** Bye-bye, LittleLaiv!

**Bot:** Bye, see you later!

오늘의 정리를 해보자면:

- 문법적으로 틀린 경우
  - 사용자가 작성한 원래 문장: "No, I doesnt"
  - 고쳐진 문장: "No, I don't."
  - 간단한 이유: 'doesn't'는 3인칭 단수 주어와 함께 쓰이는 표현이다.
- 부자연스러운 표현 사용
  - 사용자가 작성한 원래 문장: "2층 오른쪽 코너에 있어요"
  - 고쳐진 문장: "They are at the right corner on the second floor."
  - 간단한 이유: 한국어로 말하는 대신 영어로 대답을 해야한다.

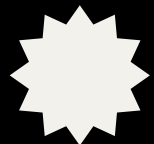
오늘 대화한 내용을 잘 기억해줬다가 다음에도 활용해봐. 수고했어!

---

### Your Input

Type your message here:





2024 DSL 기업연계 프로젝트



**LA.IV DATA**  
라이브데이터

감사합니다.