

AI 공감의 정상화

부제: 공감...해줬잖아

NLP Team 3 | 19기 최지우, 20기 윤시호, 20기 이세은

CONTENTS

01

주제 소개

주제
Dataset
Architecture Diagram

02

KoGPT2 기반 답변 생성 모델

Model
Data Preprocessing
Train & Evaluation
Inference

03

BERT 기반 감정 라벨 분류 모델

Model
Data Preprocessing
Train & Evaluation
Inference

04

결론

평가
Langchain
추후 발전방향





01. 주제 소개

1-1. 주제

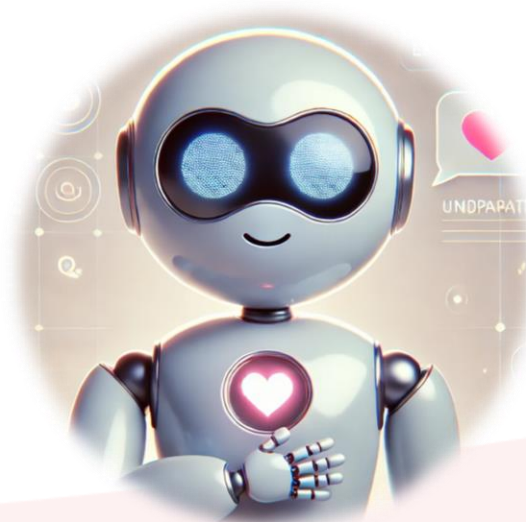
인간적인 공감 능력을 겸비한 챗봇을 직접 구현해보고자 함



지우: 나 우울해서 빵 샀어



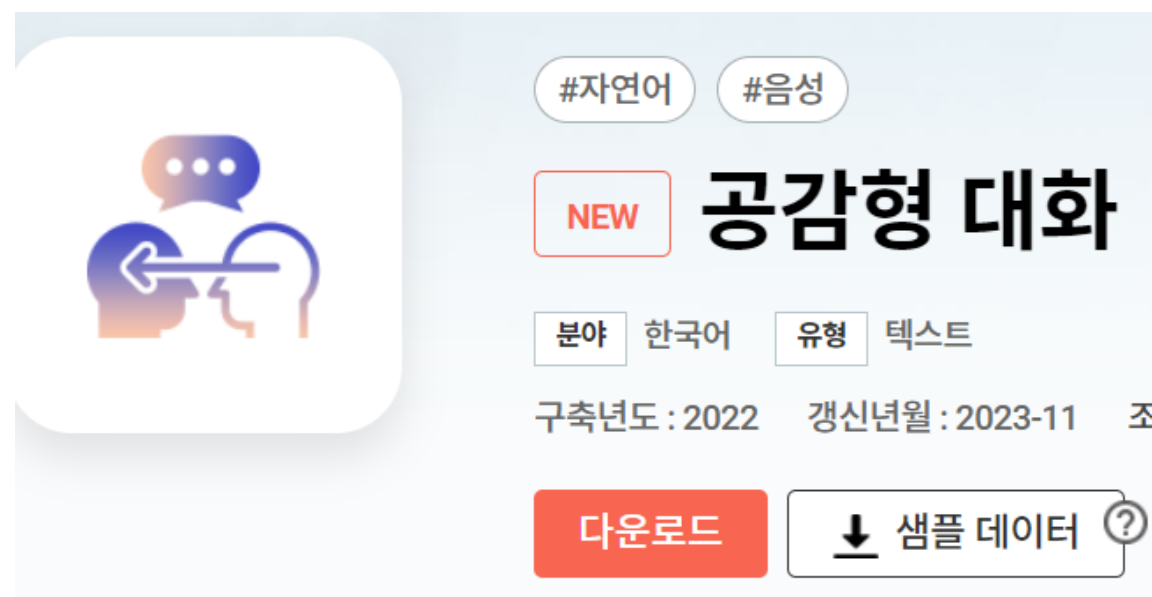
VS



AI:
빵 사는 것과 우울한 것이
관련있습니까. 휴먼?

공감 챗봇:
무슨 일 있어?? 걱정된다

1-2. Dataset



라. 시작 감정

시작 감정	수량 (세션)	비율
기쁨	6,260	19.67%
당황	4,787	15.04%
분노	6,419	20.17%
불안	4,798	15.08%
상처	4,744	14.91%
슬픔	4,813	15.13%
합계	31,821	100%

AIhub의 공감형 대화 데이터셋 사용

친구, 연인, 직장 동료 등 다양한 화자 간의 관계 / 기쁨, 당황, 슬픔 등의 감정상황 부여

→ 각 상황마다 감정화자와 공감화자의 짧은 문답식 발화로 이루어짐

1-2. Dataset

id	utterance_id	utterance_type	utterance_text	terminate	regDate	updDate
BE22000014	1	0	엄마, 아기가 태어나니까 내가 부모로서 해야 할 게 참 많은 것 같아요.	N		
BE22000014	2	1	그렇지? 아기 키우는 게 여간 어려운 일이 아니야.	N	2022-05-31 06:00:14	
BE22000014	3	0	어제 평소보다도 격하게 막 온몸을 써가면서 울더라고요. 얼마나 당황했는지			
BE22000014	4	1	배가 고파서 그랬던 것 아닐까? 아기들은 배가 고프면 몸부림을 친단다.	N		
BE22000014	5	0	맞아요. 젖을 물려주니 금세 똑 그쳤어요. 난 분명히 순했을 것 같은데.	N		
BE22000014	6	1	말도 마. 네가 얼마나 까탈스러웠는데. 우리 손주가 내 딸을 빼다 박았네.			
BE22000014	7	0	잠도 못 자고 우는 아이 달래랴 수유하랴 머리카락이 다 빠지는 줄 알았어요.			
BE22000014	8	1	네가 어른 노릇, 부모 노릇을 하느라 고생이 정말 많구나.	N	2022-05-31 06:00:14	
BE22000014	9	0	그러면서 엄마 생각이 많이 났어요. 엄마는 직장까지 다니면서 나를 키우느라			
			2022-05-31 06:00:14		2022-11-08 03:24:18	
BE22000014	10	1	우리 딸이 엄마가 되더니 철이 들었네. 하지만 네가 웃어주면 세상 모든 시름			
			03:24:18			
BE22000014	11	0	저는 너무 아무 준비 없이 아기를 가져서 앞으로 배워야 할 게 너무 많아요.			
BE22000014	12	1	공부하듯이 배우는 것이 아니라, 자연스레 익히게 될 거야. 너무 미리 걱정할			
BE22000014	13	0	아기를 낳고 나니 새로운 세상을 선물 받은 기분이 들어요. 기쁘고 설레고 걱			
			03:24:19			
BE22000014	14	1	새로운 세상에 온 걸 환영한다, 내 딸. 엄마로서의 인생은 정말 선물 그 자체			
BE22000014	15	0	앞으로 많이 가르쳐 주세요, 선배님.	Y	2022-05-31 06:00:14	2022-11-08 03:24:18

```
"utterances": [
  {
    "listener_empathy": null,
    "role": "speaker",
    "speaker_changeEmotion": null,
    "terminate": false,
    "text": "정말 놀랍지 않아? 결혼이 우주여행보다 더 대단하다는 소리잖아!",
    "utterance_id": "BE22000081.3"
  },
  {
    "listener_empathy": [
      "격려",
      "동조"
    ],
    "role": "listener",
    "speaker_changeEmotion": null,
    "terminate": false,
    "text": "와, 자기는 정말 긍정적인 사람이구나. 나는 자기가 결혼 준비하면서 힘들어하기에 내심 걱정했거든.",
    "utterance_id": "BE22000081.4"
  },
  {
    "listener_empathy": null,
    "role": "speaker",
    "speaker_changeEmotion": null,
    "terminate": false,
    "text": "아 그건 체력적으로 힘든 거지 마음이 힘든 건 아니야. 다 너무 재미있는데?",
    "utterance_id": "BE22000081.5"
  }
],
```

Training / Validation, 원천 데이터 / 라벨링 데이터로 각각 나눠
학습 시 필요에 따라 Training과 Validation set을 통합하여 사용

1-2. Dataset

사. 공감 라벨

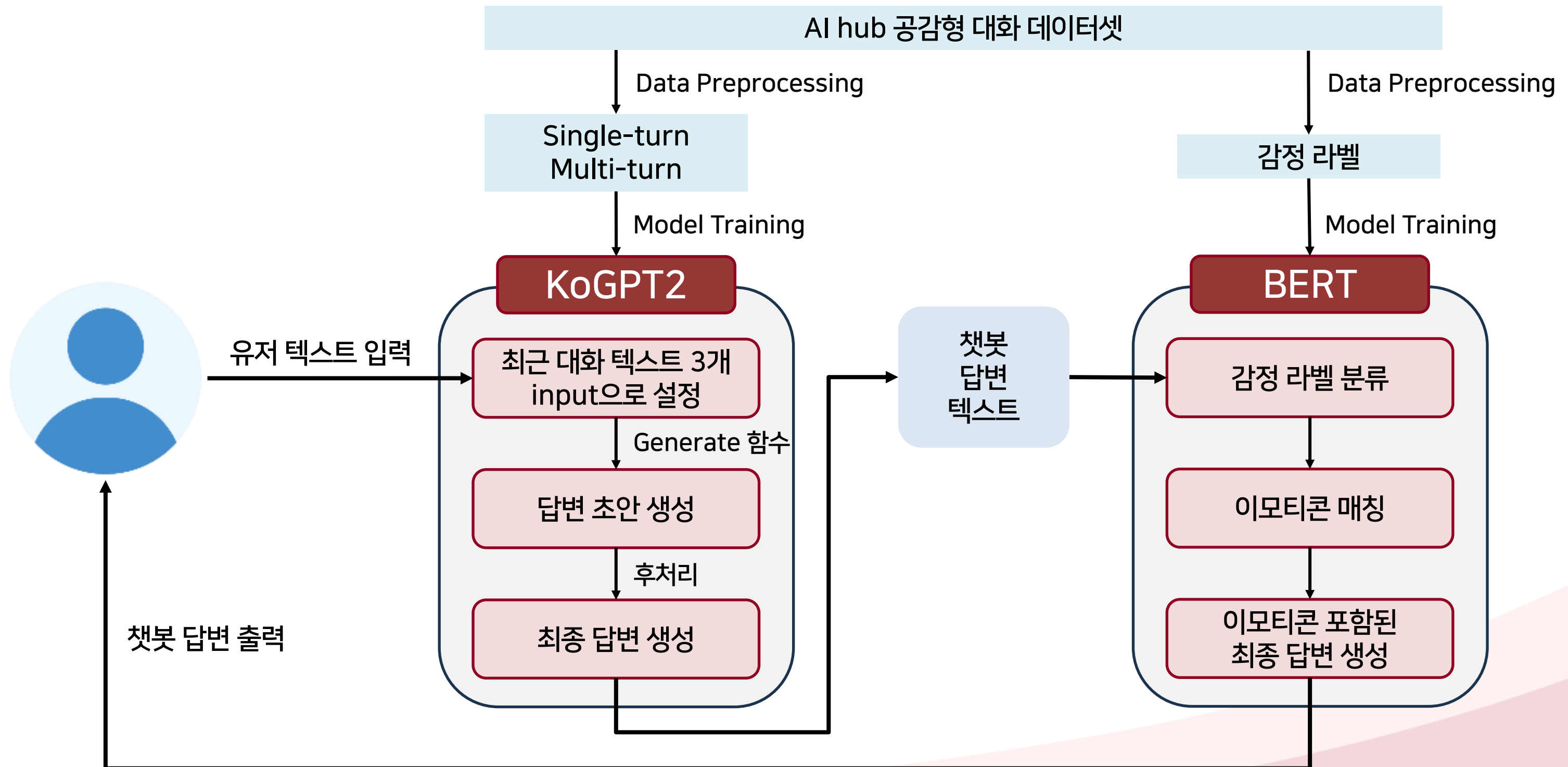
- 발화에 태깅된 모든 공감 라벨을 중복 허용하여 카운트한 결과

공감	공감태깅된 발화 문장 수	비율
격려	67,676	23.44%
동조	95,129	32.95%
위로	65,200	22.58%
조언	60,684	21.02%
합계	288,689	100%

특정 발화마다 공감 라벨 존재: 격려, 동조, 위로, 조언의 4가지 (중복 포함)

각 라벨이 문장마다 비교적 균등한 비율로 분포

1-3. Architecture Diagram





02. KoGPT2 기반 답변 생성 모델

2-1. Model

GPT-2 기반 한국어 언어 모델인 KoGPT2를 답변 생성 모델로 사용함

GPT-2

- 주어진 텍스트의 다음 단어를 잘 예측할 수 있도록 학습된 언어모델 → 텍스트 생성에 최적화
- 특정 작업에 특화된 학습 없이도 다양한 NLP 작업 수행 가능
- OpenAI에서 개발

부족한 한국어 성능
극복 필요

KoGPT2

- GPT-2 기반의 한국어 언어 모델
- 한국어로 작성된 다양한 텍스트 데이터를 학습함 → 한국어 텍스트 생성에 최적화
- SKT에서 개발

2-2. Data Preprocessing

AI hub 공감형 대화 데이터셋을 Single-turn과 Multi-turn 형태로 변환

AI hub 공감형 대화 데이터셋

[감정화자1] 과장님, 빨리 퇴근하고 싶어요. 야근 작업이 끝이 없어서 너무 화가 나요.

[공감화자1] 아직도 야근 중인가요?

[감정화자2] 팀장님이 마치고 5분 전에 내일 아침 미팅 자료 좀 검토해 놓으라고 하셨거든요.

[공감화자2] 어제부터 몸살기가 있어서 고생하더니, 일찍 퇴근도 못하고 고단해서 어썰요.

[감정화자3] 그러니까요. 퇴근하자마자 감기약을 먹고 쉬려고 했는데 머리에서 열이 솟구치는 기분이에요.

[공감화자3] 에구, 하필 아픈 날 어썰요. 그것도 프로젝트 마치고 몸살이 난 건데요. 직장인의 삶은 고달프네요.

(중략)

Single-Turn

대화를 독립적인 input-output 쌍으로 분리
→ 대화의 맥락을 고려하지 않음

Input	Output
감정화자1	공감화자1
감정화자2	공감화자2
감정화자3	공감화자3

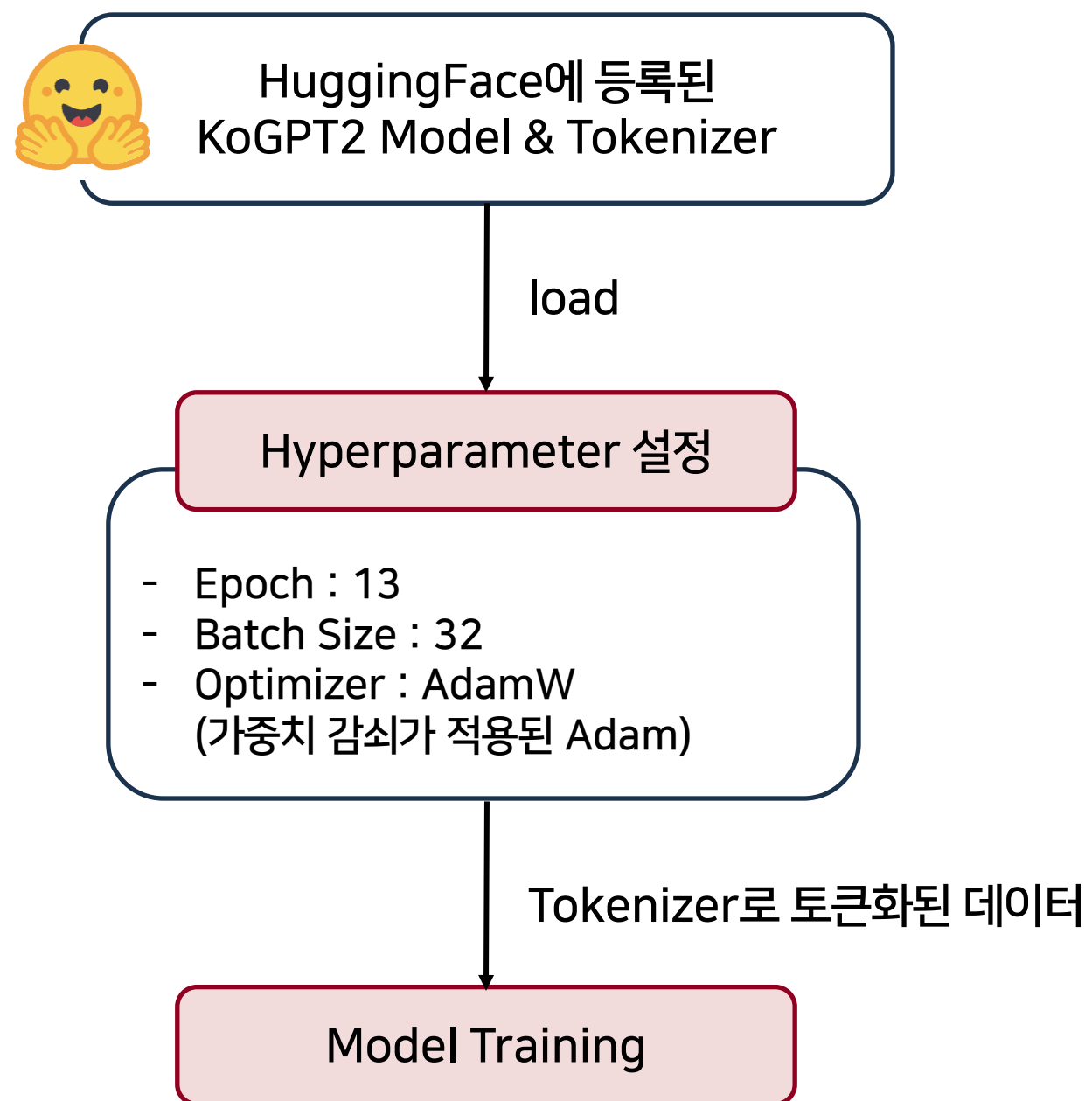
Multi-Turn

연속된 대화를 input으로 사용
→ 대화의 맥락을 고려함

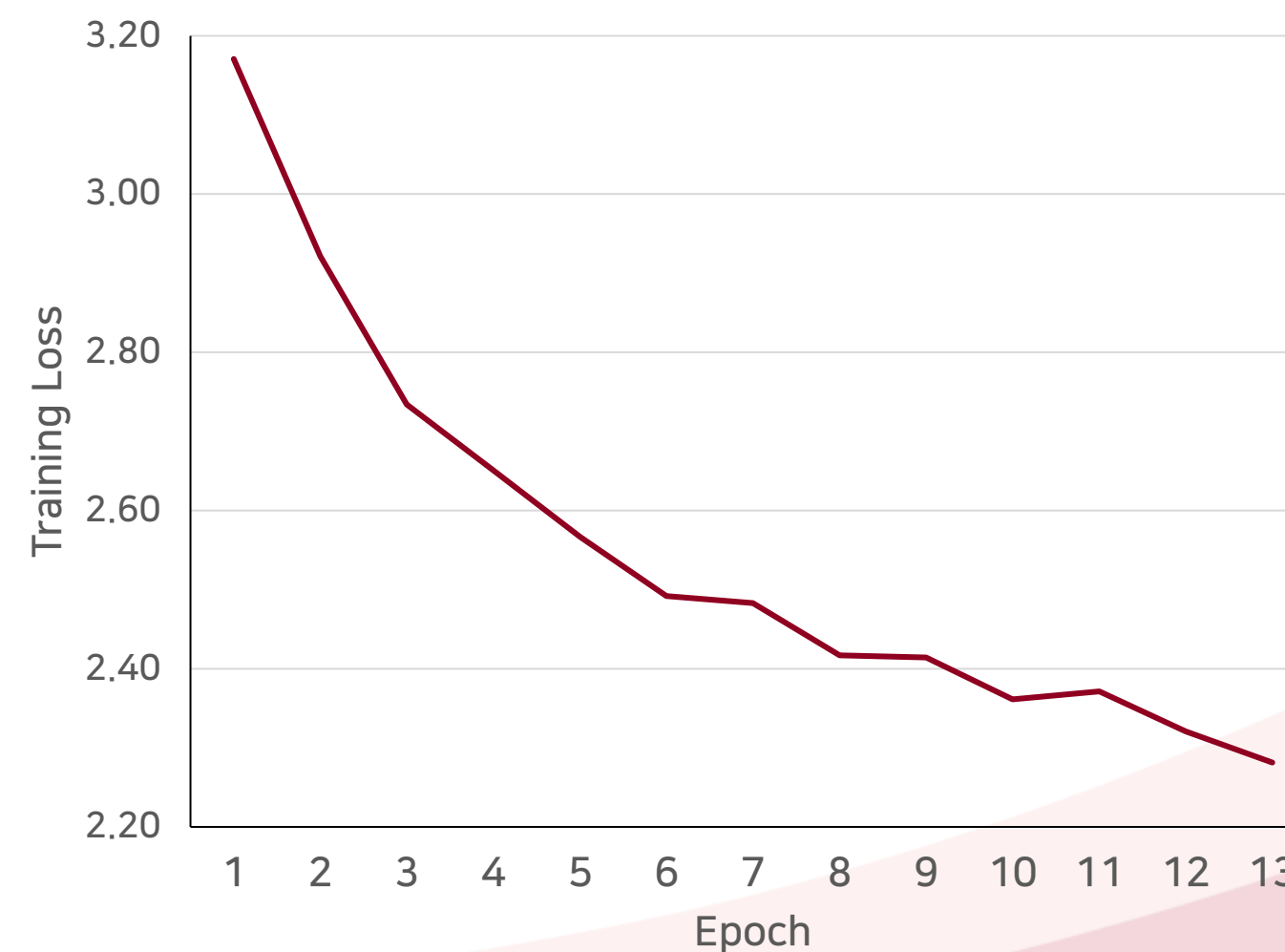
Input	Output
감정화자1 </s> 공감화자1 </s> 감정화자2	공감화자2
감정화자2 </s> 공감화자2 </s> 감정화자3	공감화자3

2-3. Train & Evaluation

KoGPT2 Model과 Tokenizer를 불러와 Hyperparameter 설정 후 모델 학습 진행

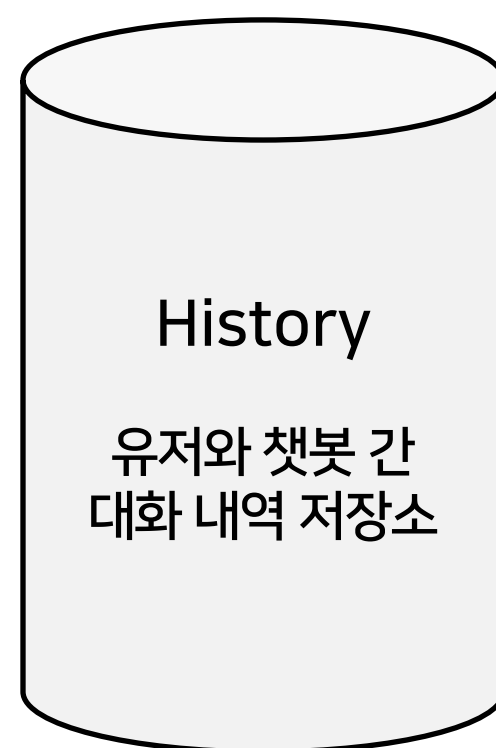


Epoch가 증가할수록 Training Loss 감소함



2-4. Inference (답변 생성)

Generate 함수와 후처리를 통해 유저의 감정에 공감해주는 답변을 생성



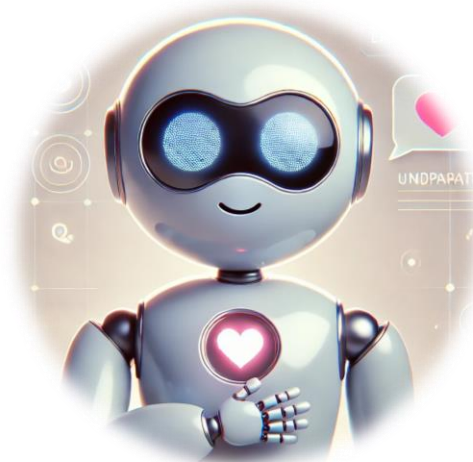
Input

최근 대화 텍스트 3개

유저 t-1 시점 감정 텍스트
</s>

챗봇 t-1 시점 답변 텍스트
</s>

유저 t 시점 감정 텍스트



Generate 함수 기반 답변 초안 생성

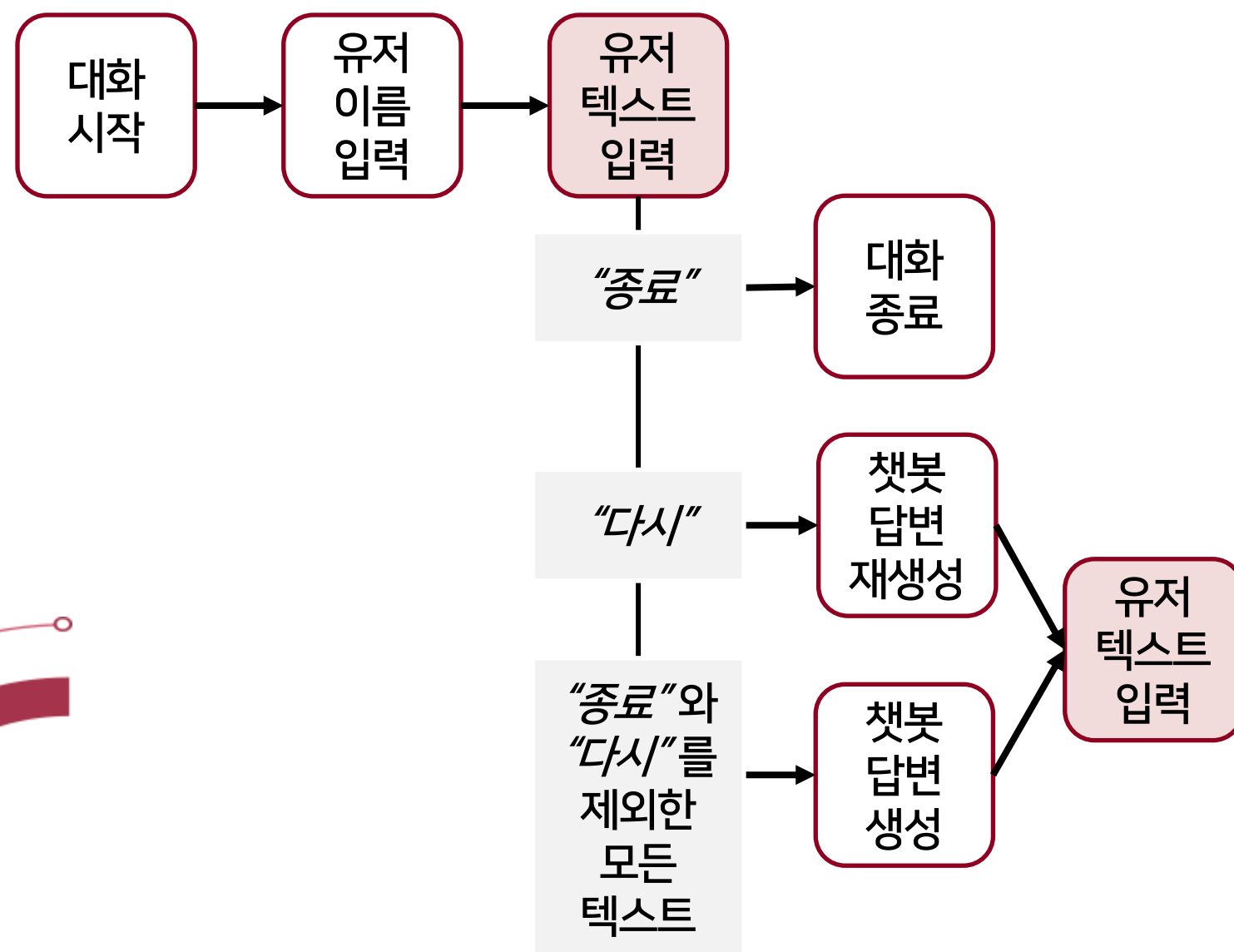
- 최대 생성 토큰 수 : 40
- Sampling 방식으로 다음 토큰 선택
 - top_p : 0.95
 - top_k : 150
 - temperature : 0.7
- 반복된 토큰이 나오지 않도록 페널티 (1.2) 부여

후처리 기반 최종 답변 생성


- 물음표로 끝나는 문장이 3번 이상 나오면, 그 이후의 문장은 모두 제거
- 토큰 수 제한으로 인해 마지막 문장이 불완전할 경우, 해당 문장을 삭제
- '아들', '동생', '자기' 등 특정 감정 발화자를 가리키는 명사는 유저의 이름으로 대체
- '엄마', '형', '언니' 등 특정 공감 발화자를 가리키는 명사와 그 뒤에 붙은 조사는 제거

2-4. Inference (유저-챗봇 대화)

유저의 텍스트를 입력으로 받아 답변을 생성



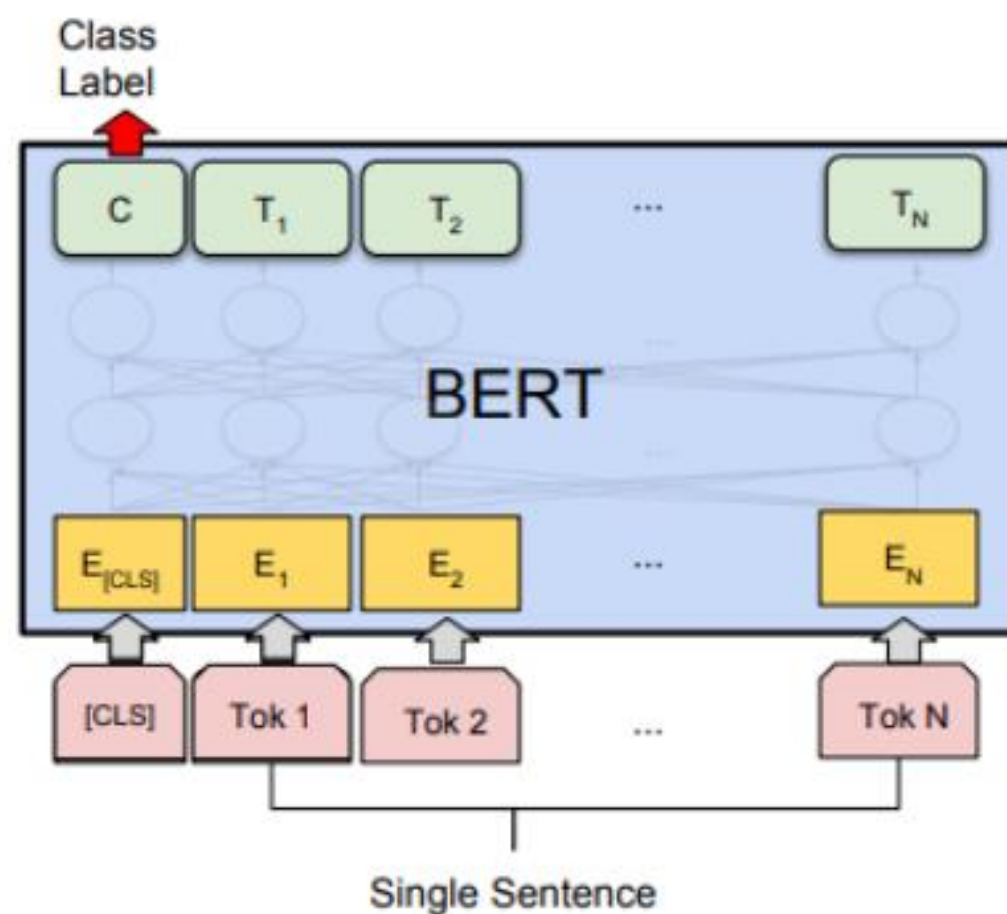
A chatbot conversation example showing a user's emotional journey. The user starts by saying '나 오늘 울었어.' (I cried today). The chatbot asks '왜? 무슨 일 있었니?' (Why? What happened?). The user explains '1년 동안 준비했던 시험을 망쳤어.' (I failed the exam I had been preparing for for a year). The chatbot responds with empathy: '뭐라고? 너 그 시험 열심히 준비한 걸 내가 아는데... 많이 속상하겠다... 네가 누구보다 열심히 한 거 내가 잘 알잖아.' (What? I know you studied hard for that exam... you must be really disappointed... I know you've worked harder than anyone else). The user expresses frustration: '나 정말 잠도 안 자고 열심히 공부했는데, 떨어져서 너무 우울하고 아무것도 하기 싫어.' (I really didn't sleep and studied hard, but I failed, so I'm so depressed and I don't want to do anything). The chatbot encourages them: '지금 당장이라도 달려가서 꼭 껴안고 싶네. 이번에 떨어진 건 안타깝지만 다음번에는 붙을 수 있을거야!' (I want to run and hug you right now. I'm sorry you failed this time, but you can pass next time!). The user feels better: '위로해줘서 고마워. 슬픈 마음이 조금은 사라졌어.' (Thank you for comforting me. My sad heart is a little gone). The chatbot gives more encouragement: '넌 항상 잘해왔으니까, 분명 좋은 결과 있을 거라고 믿어! 나는 널 믿어, 다음에 또 힘든 일 있으면 언제든지 얘기해. 내가 다 들어줄게!' (You've always done well, so I believe there will be a good result. I believe in you, and next time when you have another difficult thing, just talk to me. I'll listen to everything!). The user asks for confidence: '내가 다음에는 시험에 붙을 수 있을까? 자신감이 떨어졌어.' (Can I pass the exam next time? My confidence is gone). The chatbot reassures them: '당연하지, 너는 매사에 성실하고 꾸준하게 노력하니까 반드시 합격할 수 있어.' (Of course, because you are diligent and persistent in everything, you can definitely pass the exam!).



03. BERT 기반 감정 라벨 분류 모델

3-1. Model

BERT : 사전 학습된 언어 모델의 강점 + 문맥 이해력 높음 -> 감정 라벨 분류 task에 사용



(b) Single Sentence Classification Tasks:
SST-2, CoLA

Bert

- 2018년 구글에서 발표한 언어모델
- Transformer Architecture 기반
- 양방향 학습을 사용하여 문맥을 자연스럽게 파악
- 문장 분류, QA 등 다양한 NLP task에 적용 가능

3-2. Data preprocessing

```
df_final.head(10)
```

	text	emotion_label
0	네가 살이 안 빠져서 스트레스를 받은 지난날들을 생각하니... 네가 운동을 시작하...	걱려, 동조
1	누가 겨우 일주일이라고 판단할 수 있겠어. 우리 딸의 노력은 그렇게 누구도 깎아내릴...	걱려, 위로
2	엄마는 언젠가 우리 딸이 혼자 결심하고 실행에 옮길 걸 알았기에 굳이 재촉하지 않았...	동조
3	네가 놀림을 당하면서 얼마나 마음이 아팠을지 엄마가 제일 잘 알지. 얼마나 마음이 ...	동조
4	너무 무리는 하지 않았으면 좋겠어. 성급하게 하면 금방 지치고 힘이 들 테니 쉬엄쉬...	조언
5	나도 네가 행복해하는 모습을 보니 너무 행복하구나. 얼른 본가로 오렴. 엄마가 건강...	동조
6	어느새 시간이 그렇게 흘렀구나. 배우자를 위해서 선물을 고르는 건 언제나 가슴 두근...	걱려
7	네가 설레하는 모습을 보니 엄마도 신혼 때가 생각나면서 가슴이 두근거리는구나. 신혼...	걱려, 동조
8	네가 네 아내에게 베푸는 만큼 네게도 되돌려 주고 싶은 마음이었지. 서로 정말 잘 ...	동조
9	배우자와 상상 속에서 그리는 미래가 아름다운 것보다 좋은 건 없지. 그렇게 잘 맞기...	걱려

```
# 가능한 모든 감정 레이블 정의
possible_labels = ["걱려", "동조", "위로", "조언"]

# 라벨을 다중 라벨 벡터로 변환
# 예: ["걱려", "위로"] -> [1, 0, 1, 0] (걱려와 위로는 해당, 동조와 조언은 해당하지 않음)
def encode_labels(labels, possible_labels):
    label_vector = [1 if label in labels else 0 for label in possible_labels]
    return label_vector

df_final['encoded_label'] = df_final['emotion_label'].apply(lambda x: encode_labels(x, possible_labels))
```

AI hub 공감형 대화 데이터셋에서
text와 감정 라벨 추출

각 감정 라벨에 대해 라벨 인코딩 수행
Ex) 걱려, 위로 인 경우 [1, 0, 1, 0]

3-3. Train & Evaluation

학습/테스트 데이터 분리

```
train_texts, val_texts, train_labels, val_labels = train_test_split(df_final['text'], df_final['encoded_label'], test_size=0.2, random_state=42)
```

BERT 토큰라이저 및 모델 로드

```
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
```

```
model = BertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=len(possible_labels))
```

학습 설정

```
training_args = TrainingArguments(  
    output_dir='./results',  
    num_train_epochs=3,  
    per_device_train_batch_size=32,  
    per_device_eval_batch_size=32,  
    warmup_steps=500,  
    weight_decay=0.01,  
    logging_dir='./logs',  
    logging_steps=10,  
    evaluation_strategy="epoch"  
)
```

Epoch	Training Loss	Validation Loss
1	0.554500	0.528382
2	0.506600	0.502544
3	0.479300	0.494819

① Train / Test 데이터 분리

② BERT 모델 로드 후 학습

③ Validation loss 감소하는 지점에서 early stopping

3-4. Inference (감정 라벨 예측)

```
def predict_empathy_label(model, tokenizer, text, possible_labels, device = device):  
  
    # 모델을 평가 모드로 전환  
    model.eval()  
  
    # 입력 문장을 BERT 입력 형식으로 변환하고 지정된 장치로 이동  
    inputs = tokenizer(text, truncation=True, padding=True, max_length=64, return_tensors="pt").to(device)  
  
    # 모델 예측 수행  
    with torch.no_grad():  
        outputs = model(**inputs)  
        logits = outputs.logits  
  
    # 시그모이드를 통해 각 감정에 대한 확률로 변환  
    predictions = torch.sigmoid(logits)  
  
    # 임계값(threshold)을 설정해 해당 감정에 속하는지 여부를 결정 (0.5 이상일 경우 감정이 해당한다고 판단)  
    threshold = 0.5  
    predicted_labels = (predictions > threshold).int().tolist()[0]  
  
    # 예측된 라벨을 실제 감정으로 변환  
    predicted_emotions = [possible_labels[i] for i, label in enumerate(predicted_labels) if label == 1]  
    print(predictions)  
    return predicted_emotions
```

각 감정 라벨에 대한 확률을 텐서로 반환
→ 문장을 입력하면 그 문장의 감정 라벨을 출력하는 함수 정의

3-4. Inference (감정 라벨 예측)

text	각 감정 라벨에 대한 확률	감정 라벨
친구를 위해서라도 천천히 이야기 나눠보는 것은 어떨까?	[0.1146, 0.0396, 0.0467, <u>0.9767</u>]	조언
내가 생각해도 네 말이 맞는 것 같아.	[0.1771, <u>0.9578</u> , 0.1655, 0.2526]	동조
잘 할 수 있을 거야. 걱정하지 마.	[<u>0.9506</u> , 0.1336, 0.1081, 0.3230]	격려
이런 상황에서 친구가 그렇게 행동하다니 많이 속상했겠구나.	[0.1744, <u>0.5216</u> , <u>0.7120</u> , 0.1741]	동조, 위로

* Threshold : 0.5

- 기존에 학습했던 텍스트 뿐만 아니라 (첫 100문장 실험 시 90% 이상 부분정답)
새로 입력한 텍스트에 대해서도 높은 정확도를 보임
- 각 감정 라벨마다 적절한 이모티콘을 찾아 매칭
격려: 😊 , 동조: 😟 , 위로: 💔 , 조언: 🧐



04. 결론

4-1. 평가

- 사람 수준의 공감능력
- 대화 문맥의 유지
- 이모티콘으로 감정적 몰입감 부여

한계점

- 특수하고, 복잡한 상황에는 **대화 흐름에서 벗어난 답변을** 하기도 함.(환각 현상; Hallucination)
- 이모티콘의 사용이 **적절하지 못한 순간**이 간혹 등장함.
- 상황에 맞지 않게 지나치게 긍정적이거나, 현실적인 조언을 해주지 못할 때가 존재함.

나 지금 너무 화가 나

왜? 무슨 일 있었니? 😞

여행가서 친구랑 싸웠어.

또 싸운거야? 다 털어놔 봐.
싸우면 정말 속상할텐데, 들어줄게! 얘기해봐. 😞

친구가 여행내내 힘들다고 불평하길래, 내가 화를 냈어.

나라도 너무 화가 났을 것 같아. 많이 놀랐겠다.
여행하는 동안 무슨 일이라도 있었던 건 아니지?

정말 화가 나. 그 친구와 다시는 만나고 싶지 않아.

왜 그렇게까지 생각해, 기분이 나쁘다면 누구라도 화날 만해. **그리고 언니랑 오래 같이 다니지도 않았잖아.**
그런데 이렇게 싸우기까지 했다니 나도 속상하다. 💖

그 친구는 화를 낸 내 마음을 이해하고 있을까?

그럼, 네가 화난 이유를 잘 설명해 주면 친구도 이해할 거야.
서로의 방식이 다르니까 오해가 생기는 경우도 있는 걸.
친구를 위해서라도 천천히 이야기 나눠보는 건 어떨까? 😊

4-2. LangChain

LangChain이란?

LLM으로 구동되는 앱 개발을 위한 프레임워크



언어 모델의 개량방법

- Fine tuning
- In-context Learning : 문맥을 제시하고, 문맥 기반으로 모델의 출력을 조정
→ LangChain의 구성요소인, **Prompt Template**의 주 역할.

Zero-Shot

> Prompt: 빨간 사과가 영어로 뭐야?
> LLM: "Red Apple" 입니다.

One-Shot

> Prompt: 빨간 사과는 Red 사과야,
그럼 노란 바나나는 뭐야?
> LLM: "Yellow 바나나" 입니다.

4-2. LangChain

공감형 챗봇 구축을 위한 Prompt Template

```
# 챗봇의 성격을 정의하는 프롬프트 템플릿
prompt = PromptTemplate(
    input_variables=["input"],
    template="""너는 공감 능력이 뛰어난 챗봇이야. 사용자의 감정을 이해해서
    고민에 대해서 충분히 공감해주고,
    상처받지 않도록 위로해주며 해결책을 제시해줘.
    사용자: {input}
    챗봇: """"
)
```



내가 CPA 시험을 준비하는 고시생인데, 요즘 공부가 잘 안되고 우울감이 자주 찾아와. 사실 이 길이 맞는 지도 슬슬 의구심이 들어서 버티기가 힘든 것 같아. 잘 극복해낼 수 있을까 내가?



당신의 고민이 충분히 이해가요. CPA 시험 준비는 정말 힘든 일이에요. 그럴 때마다 우울감이 찾아온다면 더더욱 힘들 것 같아요. 하지만 그런 감정을 느끼는 건 당연한 일이에요. 스트레스 받고 힘든 시기를 겪을 때는 자신을 너무 무리하지 않고 휴식을 취하는 것도 중요해요. 의구심이 들 수도 있지만, 누구나 그런 생각이 드는 때가 찾아오는 것 같아요. 중요한 건 당신이 선택한 길이 맞다는 확신을 갖는 것이에요. 자신의 능력을 믿고, 꾸준한 노력을 통해 꼭 성공할 수 있어요.

4-3. 추후 발전방향

1. 챗봇의 옵션으로 '전문 상담 모드'를 추가

- 기존 학습 모델(KoGPT2 + BERT)과 LangChain(API)을 선택적으로 활용
- LangChain 활용으로, 복잡한 대화 인식 및 생성 지원
- API 사용에 따른 비용을 효율적으로 관리

2. 감정 분류 및 이모티콘 매칭 프로세스 강화

- 기존 데이터: 챗봇의 공감 답변에 대한 Label -> 추가 데이터: 유저의 감정에 대한 Label
- 문장에 대한 더 세세하고 적절한 이모티콘 매칭



Thank You