모델 기술서	
참가팀	포항항
모델명	행집욕부! 행복에 집중하기, 욕심 부리지 말기! 아자아자 화이팅!
작성 일자	2024.08.30
모델 깃허브 링크	https://github.com/oneonlee/KR-Conversation-Inference_Refined

1. 라이브러리 및 데이터

본 모델이 목표로 하는 '대화 맥락 추론' 과제는 다중 선택형 문제로, 모델의 설계 과정은 주어진 발화와 대화 내용에 따라 여러 추론문 중 하나의 적절한 문장을 선택하는 능력에 중점을 두었다. 본 모델은 주어진 발화에 대한 추론을 성공적으로 수행하기 위해 최첨단 언어 모델들을 사용하고 있으며, 이를 효과적으로 구현하고 최적화하기 위해 다양한 라이브러리를 활용하였다.

1) 주요 스펙 및 라이브러리

o Python 3.9.19 및 Ubuntu 20.04.5 LTS
- 기본 개발 언어로 Python 3.9.19 버전과 Ubuntu 20.04.5 LTS 운영체제를 사용하였음.

o GPU 환경
- 모델의 훈련에는 A100-PCIe-40GB GPU 1대를 사용하였으며, 모델의 추론에는 RTX 4090 (24GB) GPU 1대를 사용하였음.

o Torch (v2.3.1) 및 BitsandBytes (v0.43.1)
- PyTorch 라이브러리를 사용해 딥러닝 모델을 설계, 학습, 추론했으며, BitsandBytes는 모델 훈련 시 사용된 저정밀 계산을 처리하기 위해 사용하였음.

o PEFT (v0.11.1)
- 메모리 사용량을 최소화하며 LLM을 효율적으로 미세 조정하는 Parameter Efficient Fine Tuning (PEFT) [1] 기법을 위해 해당 라이브러리를 사용했으며, LoRA [2] 및 QLoRA [3]와 같은 기법을 통해, 파라미터의 일부만을 조정하는 방법을 채택하였음.

o Transformers (v4.41.1)
- 해당 라이브러리는 Hugging Face에서 제공하는 딥러닝 라이브러리로, 사전 학습된 대형 언어 모델을 불러와 손쉽게 미세 조정을 수행하였음.

o Pandas (v2.2.2) 및 Numpy (v2.0.0)
- 해당 라이브러리들은 데이터 처리 및 분석에 널리 사용되는 라이브러리로 데이터셋의 전처리와 통계 분석에 중요하게 사용되었음.

o Datasets (v2.20.0)
- 데이터셋 관리를 용이하게 하기 위해 Hugging Face의 'Datasets' 라이브러리를 사용하였으며, 모델의 입력 형식에 맞는 전처리를 수행하였음.

o Matplotlib (v3.9.2) 및 WordCloud (v1.9.3)

- 데이터 분석의 시각화를 위해 사용하였으며 데이터의 분포와 특성을 시각적으로 분석했음.

2) 데이터

본 모델은 국립국어원 인공지능(AI) 말평에서 제공한 대화 맥락 추론 말뭉치 데이터를 기반으로 제작되었다. 데이터는 1,514건의 대화 형태로 이루어져 있으며, 각 대화는 문장 단위의 발화로 구성되어 있다. 모든 대화는 '원인', '후행 사건', '전제 조건', '내적 동기', '감정 반응' 등 5가지 유형으로 분류된 추론문과 짝지어져 있으며, 각 대화 내 발화는 다섯 가지의 추론 유형에 대응하는 추론문을 예측하는데 사용되었다. **모델의 학습에는 분할 제공된 데이터의 훈련 데이터 758건만을 사용**하였으며, **검증 데이터 151건과 시험 데이터 605건은 모델 학습 과정에 포함하지 않았다**. <그림 1>은 제공되는 데이터셋의 예시이다.

Original Dataset

■ 대화(conversation)
- 화자1: "안녕하세용~~~~"
- 화자1: "name2님은 혹시 이상형이 어떻게 되시나요???"
- 화자2: "안녕하세요"
- 화자2: "반갑습니다"
- 화자2: "저는 어른스러운 사람을 선호하는 것 같아요ㅎㅎㅎ"
- 화자2: "name1님은요????"
- 화자1: **"오 저도 나이보다는 생각이 좀더 성숙하고 어른스러운 사람을 좋아하는거 같아요"**
- 화자1: **"생각이 좀 미성숙하면 대화가 잘 안돼서 힘들었달까....."**
- 화자2: "맞아요 나이만 많다고 해서 어른스럽지는 않더라고요ㅜㅜ"
- 화자1: "맞아요 맞아요... 나이보다는 내면의 성숙함이 더 중요하죠....."
- 화자2: "그쵸"
- 화자1: "그리고 저는 한 번 말한 건 꼭 지키는 사람도 선호해요"
- 화자1: "어떤 문제로 싸우더라도 그 문제로 다시는 안싸우게 되는.....!"
- 화자2: "아 그거 좋네요"
- 화자1: "같은 문제로 계속해서 싸우면 도돌이표 같고 반복되는 느낌같아서"
- 화자1: "너무 시간낭비처럼 느껴지더라구요"
- 화자2: "맞아요 그렇게 되면 감정 소모가 의미 없게 느껴지죠"
- 화자2: " 비슷한 맥락으로"
- 화자2: "화가 나거나 급한 상황에서도 말을 신중하게 하는 사람이 좋아요"
- 화자1: "오... 그쵸 그런 상황에서는 서로 상처받기 쉽죠"
- 화자1: "저는 욕 하는 사람들도 별로 좋아하지는 않아요.."
- 화자1: "장난으로라도 욕을 쓰는건 싫더라구요"
- 화자2: "진짜ㅋㅋㅋㅋㅋㅋㅋ 나이만 많고 욕만 자주 하는 사람들 보면 좀 한심해요..ㅋㅋㄲㅋㅋ"
- 화자1: "맞아요..... 욕 자체가 자신을 낮추는 행동인데... 사람들이 너무 많이 쓰더라고요....."

■ 추론문 유형(category)
- "원인"

■ 추론문 후보(inference)
- A: "화자1은 깊이 있고 높은 수준의 대화를 중요하게 생각한다."
- B: "화자1은 상대방의 나이를 가장 중요한 요소로 본다."
- C: "화자1은 이성의 성숙도보다는 나이를 더 중요하게 여긴다."

<그림 1> 대화 맥락 추론 말뭉치 데이터 예
(발화 중 **굵게 밑줄**이 쳐져 있는 발화는 추론문 선택 시 참고해야 하는 특정 대상 발화를 나타냄)

2. 데이터 전처리

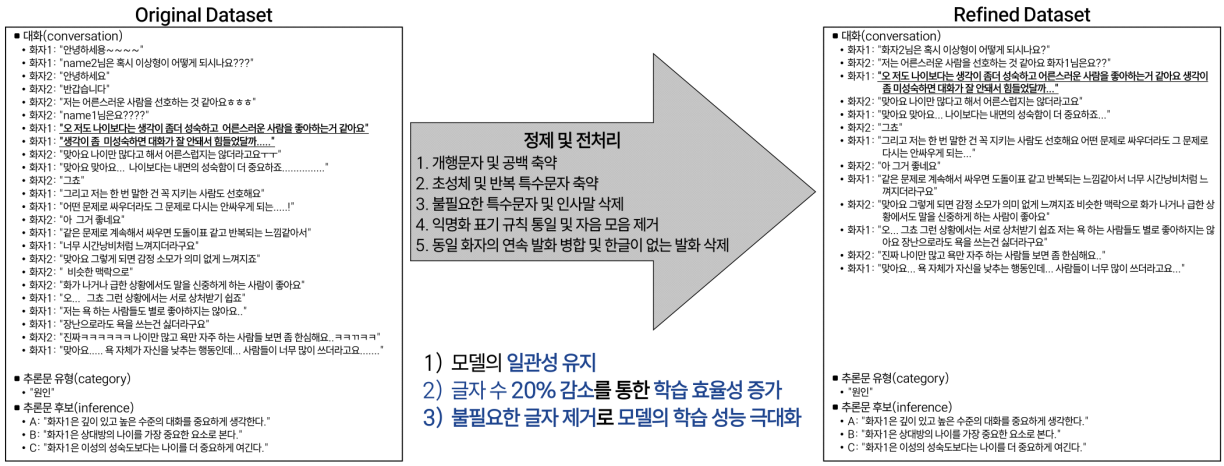
데이터 정제 및 전처리는 원시 데이터에서 발생할 수 있는 오염 요소를 제거하고, 학습 모델의 특성과 목적에 맞게 최적화하는 단계이다. 구체적인 데이터 전처리 과정은 모델 성능의 기초를 다지는 중요한 단계이며, 효과적인 모델 학습을 위해선 데이터셋에 대한 철저한 정제 및 전처리 과정이 필요하다. 본 모델의 학습과 추론에는 정제 및 전처리를 통해 **최적화된 데이터를 활용하여 대화 맥락을 보다 명확하게 추론**할 수 있도록 구성하였으며, **프롬프트 수립 과정을 통해 모델이 대화 맥락 속에서 보다 정확한 추론을 하도록 유도**하였다.

1) 정제 및 전처리의 목적

성능을 높이는 데에 중요한 것은 **모델이 유의미한 문맥을 효과적으로 포착하게 하는 것**이다. 데이터 전처리 시에 **텍스트 정제**, **불필요한 발화 제거**, **발화 병합** 등을 주로 수행하여 **데이터의 일관성을 유지**하면서도 **입력 데이터의 길이를 감소**시켜 **모델의 학습 성능을 극대화**하는 것을 목적으로 하였다.

2) 정제 및 전처리 단계

데이터 정제 및 전처리는 <그림 2>와 같은 단계로 수행하였으며, 각 과정의 세부 구현 코드는 모델 깃허브 리포지토리 내 run/refine.py 파일에 포함되어 있다.



<그림 2> 데이터 정제 및 전처리 과정

① 개행문자 및 공백 축약

- 제공되는 데이터를 확인하였을 때 개행 문자가 많거나 여러 공백이 포함되어 있는 경우가 있었는데, 이를 하나의 공백으로 축약하여 대화 내용의 가독성을 증가시킴.

② 초성체 및 반복 특수문자 축약

- 초성체('ㅇㅇㅇ' 등)나 반복된 특수문자('.....', '?????')는 그들의 의미를 명확히 하고자 축약 또는 변환하여 처리함. 예를 들어, 'ㅇㅇㅇ'은 '응'으로 변환하고, 반복된 특수문자는 의미가 명확해지도록 축약하여 '...', '??'로 변환함.

③ 불필요한 특수문자 및 인사말 삭제

- 문장에 포함된 의미 없는 특수문자('~', '_', '/', 등)와 대화 맥락과 관련 없는 인사말('안녕하세요', '반갑습니당')을 제거하여 대화의 핵심 내용에 집중할 수 있도록 처리함.

④ 익명화 표기 규칙 통일 및 자음 모음 제거

- <그림 1>에서 확인할 수 있듯이, 제공된 기존 데이터셋에는 익명화 표기 규칙이 '[name1]', '화자1'과 같이 혼재되어 있음.
- 이를 통일화하여 '[name1]'과 같은 부분을 '화자1', '화자2'로 변경함.
- 또한, 자음 및 모음만 포함된 텍스트 중 대화 내용과 관련 없는 텍스트(예: 'ㅠㅡㅠ', 'ㅋㄱㅋㄲㅋ')는 제거하여 대화의 일관성을 유지함.

⑤ 동일 화자의 연속 발화 병합 및 한글이 없는 발화 삭제

- 대화의 흐름을 더 명확히 하도록 처리하고자, 동일 화자의 연속 발화를 병합함.
- 또한, 발화에 한글이 포함되지 않고, 특수문자나 숫자만 남아있는 발화는 삭제함.

3) 프롬프팅 전략 적용

효과적인 데이터 처리를 위해 프롬프팅(prompting) 전략을 추가하여 모델의 성능을 극대화하였다. 프롬프팅 전략은 다양한 프롬프트 템플릿(prompt template)을 사용하여 모델이 보다 정확한 대화 맥락을 추론 할 수 있도록 도왔으며, <그림 3>은 학습에 사용된 다양한 프롬프팅 전략의 예시를 보여준다.

⓪ 기본 프롬프트 템플릿

- 기준 모델의 학습과 평가를 재현하기 위한 [1]코드에 사용된 기본 프롬프트 템플릿
- 기본 시스템 프롬프트(system prompt)와 대화, 추론문 유형을 담은 질문과 추론문을 제공

① CustomRefDataset 프롬프트 템플릿

- 정확한 문제 풀이를 위해, 기본 프롬프트 템플릿(⓪)에 **특정 발화(발화 참조)를 추가**
- 프롬프트에 정보를 명시했을 때 모델의 성능이 증가한다는 연구 결과 [4]에 따라, 모델의 입력으로 하는 질문에 **특정 발화(발화 참조)를 명시**

② CustomRefOfficialTermDataset 프롬프트 템플릿

- CustomRefDataset(①)에서 사용한 질문 속 **추론문 유형의 비공식 표현을 공식 표현으로 대체**
- 추론문 유형의 공식 표현은 <표 1> 참조

③ CustomRefDefinitionDataset 프롬프트 템플릿

- CustomRefDataset(①)에서 사용한 질문 속 **추론문 유형의 비공식 표현을 정의로 대체**
- 추론문 유형의 정의는 <표 1> 참조

④ CustomRefInstructionDataset 프롬프트 템플릿

- 명령문 (instruction) 형태의 프롬프트를 명시했을 때 모델의 성능이 증가한 연구 결과 [5]에 따라 기본 프롬프트 템플릿(⓪)에서 사용된 질문 대신 **명령문을 대화 이전에 오도록 배치**

⑤ SystemRefOfficialTermDataset 프롬프트 템플릿

- CustomRefOfficialTermDataset(②)의 시스템 프롬프트 대신, **프롬프트 엔지니어링 기법 [6]을 통해 설계한 구체화 된 시스템 프롬프트 사용**

1) https://github.com/teddysum/Korean_CCI_2024

Prompt (Baseline)

```
<system_prompt>You are a helpful AI assistant. Please answer the user's questions kindly. 당신은 유능한 AI 어시스턴트 입니다. 사용자의 질문에 대해 친절하게 답변해주세요.</system_prompt>

<user_prompt>
[Conversation]
…(생략)…
화자2: 저는 어른스러운 사람을 선호하는 것 같아요ㅎㅎㅎ
화자2: name1님은요????
화자1: 오 저도 나이보다는 생각이 좀더 성숙하고 어른스러운 사람을 좋아하는거 같아요
화자1: 생각이 좀 미성숙하면 대화가 잘 안돼서 힘들었달까.....
…(생략)…

[Question]
위 대화의 원인으로 올바른 지문은?

[Option]
A. 화자1은 깊이 있고 높은 수준의 대화를 중요하게 생각한다.
B. 화자1은 상대방의 나이를 가장 중요한 요소로 본다.
C. 화자1은 이성의 성숙도보다는 나이를 더 중요하게 여긴다.
</user_prompt>
```

Prompt (`CustomRefDataset`)

```
<system_prompt>You are a helpful AI assistant. Please answer the user's questions kindly. 당신은 유능한 AI 어시스턴트 입니다. 사용자의 질문에 대해 친절하게 답변해주세요.</system_prompt>

<user_prompt>
[Conversation]
…(생략)…
화자2: 저는 어른스러운 사람을 선호하는 것 같아요ㅎㅎㅎ
화자2: name1님은요????
화자1: 오 저도 나이보다는 생각이 좀더 성숙하고 어른스러운 사람을 좋아하는거 같아요
화자1: 생각이 좀 미성숙하면 대화가 잘 안돼서 힘들었달까.....
…(생략)…

[Utterance References]
화자1 : 오 저도 나이보다는 생각이 좀더 성숙하고 어른스러운 사람을 좋아하는거 같아요
화자1 : 생각이 좀 미성숙하면 대화가 잘 안돼서 힘들었달까.....

[Question]
위 대화의 특정 발화인 'Utterance References'에 대한 원인으로 올바른 지문은?

[Option]
A. 화자1은 깊이 있고 높은 수준의 대화를 중요하게 생각한다.
B. 화자1은 상대방의 나이를 가장 중요한 요소로 본다.
C. 화자1은 이성의 성숙도보다는 나이를 더 중요하게 여긴다.
</user_prompt>
```

Prompt (`CustomRefOfficialTermDataset`)

```
<system_prompt>You are a helpful AI assistant. Please answer the user's questions kindly. 당신은 유능한 AI 어시스턴트 입니다. 사용자의 질문에 대해 친절하게 답변해주세요.</system_prompt>

<user_prompt>
[Conversation]
…(생략)…
화자2: 저는 어른스러운 사람을 선호하는 것 같아요ㅎㅎㅎ
화자2: name1님은요????
화자1: 오 저도 나이보다는 생각이 좀더 성숙하고 어른스러운 사람을 좋아하는거 같아요
화자1: 생각이 좀 미성숙하면 대화가 잘 안돼서 힘들었달까.....
…(생략)…

[Utterance References]
화자1 : 오 저도 나이보다는 생각이 좀더 성숙하고 어른스러운 사람을 좋아하는거 같아요
화자1 : 생각이 좀 미성숙하면 대화가 잘 안돼서 힘들었달까.....

[Question]
위 대화의 특정 발화인 'Utterance References'에 대한 원인(cause)으로 올바른 지문은?

[Option]
A. 화자1은 깊이 있고 높은 수준의 대화를 중요하게 생각한다.
B. 화자1은 상대방의 나이를 가장 중요한 요소로 본다.
C. 화자1은 이성의 성숙도보다는 나이를 더 중요하게 여긴다.
</user_prompt>
```

Prompt (`CustomRefDefinitionDataset`)

```
<system_prompt>You are a helpful AI assistant. Please answer the user's questions kindly. 당신은 유능한 AI 어시스턴트 입니다. 사용자의 질문에 대해 친절하게 답변해주세요.</system_prompt>

<user_prompt>
[Conversation]
…(생략)…
화자2: 저는 어른스러운 사람을 선호하는 것 같아요ㅎㅎㅎ
화자2: name1님은요????
화자1: 오 저도 나이보다는 생각이 좀더 성숙하고 어른스러운 사람을 좋아하는거 같아요
화자1: 생각이 좀 미성숙하면 대화가 잘 안돼서 힘들었달까.....
…(생략)…

[Utterance References]
화자1 : 오 저도 나이보다는 생각이 좀더 성숙하고 어른스러운 사람을 좋아하는거 같아요
화자1 : 생각이 좀 미성숙하면 대화가 잘 안돼서 힘들었달까.....

[Question]
주어진 대화의 내용과 문맥으로 미루어 보았을 때, 특정 발화인 'Utterance References'에 대해 대화의 사건을 유발하는 사건을 가장 잘 설명하는 문장은 무엇인가?

[Option]
A. 화자1은 깊이 있고 높은 수준의 대화를 중요하게 생각한다.
B. 화자1은 상대방의 나이를 가장 중요한 요소로 본다.
C. 화자1은 이성의 성숙도보다는 나이를 더 중요하게 여긴다.
</user_prompt>
```

Prompt (`CustomRefInstructionDataset`)

```
<system_prompt>You are a helpful AI assistant. Please answer the user's questions kindly. 당신은 유능한 AI 어시스턴트 입니다. 사용자의 질문에 대해 친절하게 답변해주세요.</system_prompt>

<user_prompt>
[Instruction]
주어진 대화의 내용과 문맥을 바탕으로, 특정 발화인 'Utterance References'에 대한 대화의 사건을 유발하는 사건을 가장 잘 설명하는 문장을 선택하라.

[Conversation]
…(생략)…
화자2: 저는 어른스러운 사람을 선호하는 것 같아요ㅎㅎㅎ
화자2: name1님은요????
화자1: 오 저도 나이보다는 생각이 좀더 성숙하고 어른스러운 사람을 좋아하는거 같아요
화자1: 생각이 좀 미성숙하면 대화가 잘 안돼서 힘들었달까.....
…(생략)…

[Utterance References]
화자1 : 오 저도 나이보다는 생각이 좀더 성숙하고 어른스러운 사람을 좋아하는거 같아요
화자1 : 생각이 좀 미성숙하면 대화가 잘 안돼서 힘들었달까.....

[Option]
A. 화자1은 깊이 있고 높은 수준의 대화를 중요하게 생각한다.
B. 화자1은 상대방의 나이를 가장 중요한 요소로 본다.
C. 화자1은 이성의 성숙도보다는 나이를 더 중요하게 여긴다.
</user_prompt>
```

Prompt (`CustomRefOfficialTermDataset`)

```
<system_prompt>당신은 단일 선택 질문(single-choice questions)에 정확하게 답변하는 데 도움을 주기 위해 훈련된 고급 언어 모델입니다.
이제부터 당신은 '대화 맥락 추론' 과제를 수행해야 합니다. 이 과제는 입력으로 주어진 '대화 (Conversation)'를 바탕으로, '특정된 대상 발화 (Utterance References)'로부터 '추론문 유형'에 가장 적절한 추론문을 선택하는 것입니다. 각 대화에 대해 세 개의 추론문 후보가 제공되며, 당신은 해당 대화를 정확하게 파악하고, 제시된 특정 대상 발화에 대해, 세 가지 추론문 중에서 추론문 유형에 가장 적합한 하나의 정답을 선택해야 합니다.

다음 지침을 주의 깊게 따르세요:

1. **대화 내용을 주의 깊게 읽으세요**: 대화의 전체적인 흐름과 구조를 이해하세요.
2. **대상 발화를 이해하세요**: 대화 중 특정된 발화가 무엇을 의미하는지 파악하세요.
3. **추론문 유형을 고려하세요**: 주어진 추론문이 '원인(cause)', '후행 사건(subsequent event)', '전제 조건(prerequisite)', '내적 동기(motivation)', '감정 반응(emotional reaction)' 중 어떤 유형인지 확인하세요.
4. **가장 적절한 추론문을 선택하세요**: 세 개의 추론문 후보('Option') 중 대상 발화에 가장 적합한 하나를 선택하세요.

항상 다음 형식으로 답변을 제공하세요:
- 선택한 추론문에 해당하는 문자 (A, B, C).

작업을 진행하세요.</system_prompt>

<user_prompt>
[Conversation]
…(생략)…

[Utterance References]
…(생략)…

[Question]
위 대화의 특정 발화인 'Utterance References'에 대한 원인(cause)으로 올바른 추론문은?

[Option]
…(생략)…
</user_prompt>
```

<그림 3> 데이터셋에 다양한 프롬프트 템플릿(prompt template)을 적용한 예시
가장 왼쪽 위의 'Baseline'은 기준 모델에 사용된 기본 프롬프트 템플릿(⓪)을 나타낸 것이며,
나머지 프롬프트 템플릿은 본 모델에 사용된 프롬프트 템플릿(①~⑤)임

<표 1> 다섯 가지 추론문 유형의 정의

추론문 유형		정의
비공식 표현	공식 표현	
원인	원인(cause)	대화의 사건을 유발하는 사건
후행사건	후행 사건(subsequent event)	대화 이후에 일어날 수 있는 사건
전제	전제 조건(prerequisite)	대화의 사건을 가능하게 하는 상태 혹은 사건
동기	내적 동기(motivation)	대화를 일으키는 '화자'의 감정이나 기본 욕구
반응	감정 반응(emotional reaction)	대화 사건에 대해 '청자'가 보일 수 있는 감정 반응

3. 데이터 분석

본 장에서는 제공된 대화 맥락 추론 말뭉치 데이터셋에 대해, 데이터의 분포와 특징을 명확히 이해하고자 탐색적 데이터 분석(EDA; Exploratory Data Analysis)을 수행한 결과를 중심으로 기술한다.

1) 워드클라우드를 통한 대화 내용 분석

<그림 4>는 데이터셋 속 대화 내용의 다양성을 시각적으로 검토하기 위해 워드클라우드를 통해 데이터를 분석한 결과이다. 워드클라우드는 대화 중 상위 빈도 단어들의 특성과 패턴을 보여주므로 대화의 주요 주제를 파악하는 데 유용하다. <그림 4>를 통해 주어진 데이터셋의 대화가 일상적인 주제로 진행되는 것을 확인하였으며, 자음이나 모음으로 이루어진 초성체의 빈도 수가 높은 것으로 보아, 데이터셋 내 대화가 음성이 아닌 문자로 진행된 것으로 추측하였다.

Word Cloud (Train & Dev Data)

Word Cloud (Test Data)

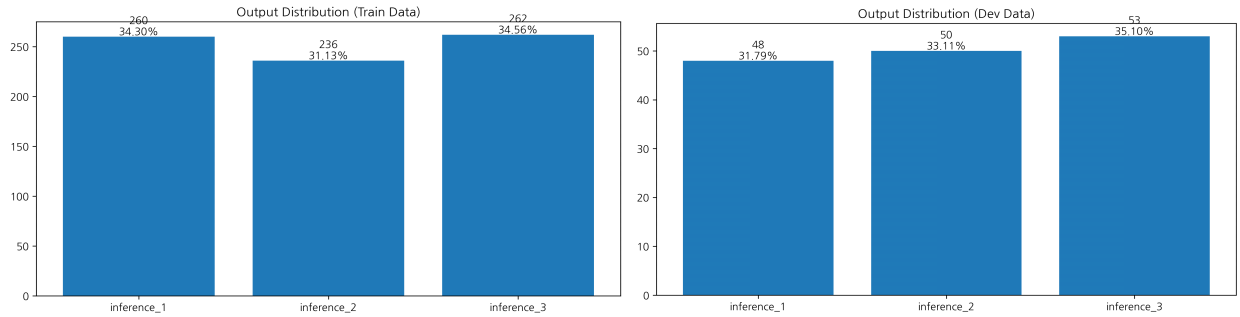
<그림 4> 대화 맥락 추론 말뭉치 데이터셋의 워드클라우드(wordcloud) 분석
(좌: 훈련 데이터와 검증 데이터를 한 번에 분석한 결과, 우: 시험 데이터를 분석한 결과)

2) 추론문 유형 및 정답 레이블의 분포

<표 2>는 각 대화마다 모델이 추론해야 하는 추론문 유형의 분포를 분석한 것이고, <그림 5>는 특정 번호가 과도하게 정답으로 선정되었는지 분석하기 위해 정답 레이블의 분포를 시각화한 것이다. 분석 결과 극단적으로는 불균형한 분포를 확인할 수 없었고 분포가 대체적으로 일정했기에, 과대표집(oversampling)이나 과소표집(undersampling) 같은 기법들을 사용하지는 않았다.

<표 2> 대화 맥락 추론 말뭉치 데이터셋의 다섯 가지 추론문 유형 분포

	원인	후행사건	전제	동기	반응	누계
훈련	161건 (21%)	118건 (16%)	161건 (21%)	156건 (21%)	162건 (21%)	758건
검증	25건 (16.5%)	51건 (34%)	25건 (16.5%)	25건 (16.5%)	25건 (16.5%)	151건
시험	117건 (19%)	133건 (22%)	115건 (19%)	124건 (21%)	116건 (19%)	605건
소계	303건 (20%)	302건 (20%)	301건 (20%)	305건 (20%)	303건 (20%)	1,514건

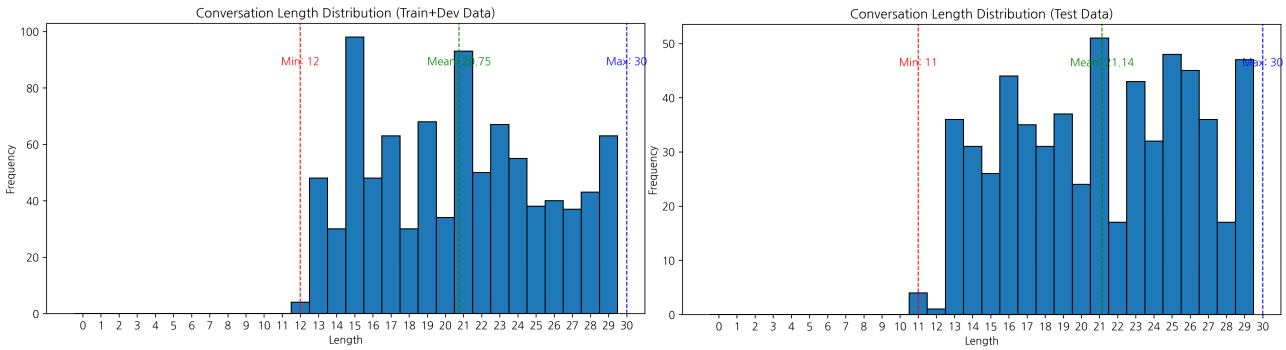


<그림 5> 정답 레이블 번호 별 분포
(좌: 훈련 데이터의 분포, 우: 검증 데이터의 분포)

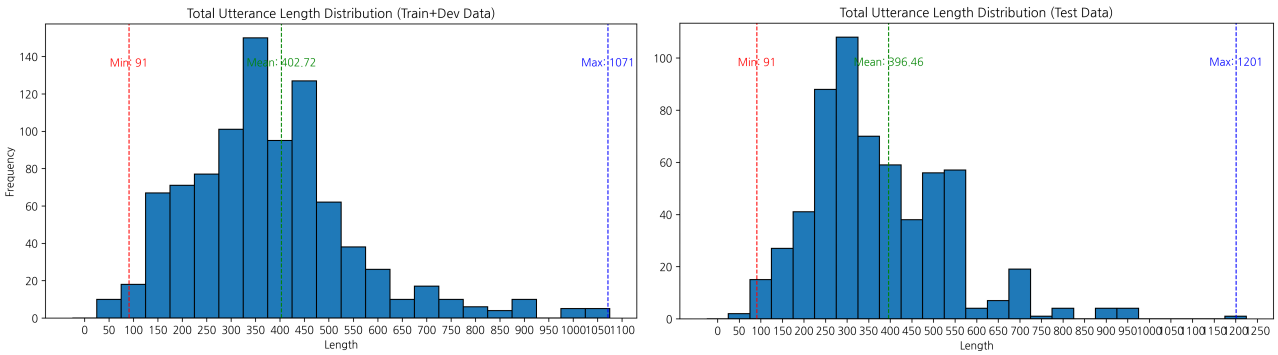
3) 대화 턴 수 분포 및 전체 대화의 길이 분포

① 제공된 원본 데이터셋의 분포 (정제 및 전처리 이전)

제공된 데이터셋은 유사한 분포를 보였다. <그림 6>은 대화를 이루는 발화가 몇 번 이루어졌는지를 의미하는 턴(turn) 수를 측정하고 분포를 시각화한 결과이다. 전체적으로 대화 턴은 최소 11턴에서 최대 30턴까지 분포하며, 각 대화는 평균적으로 약 21개의 발화를 포함하고 있는 것으로 확인되었다. <그림 7>은 전체 대화의 길이 분포를 시각화한 결과로, 최소 91자에서 최대 1,201자까지 다양한 분포를 이루었다. 각 대화는 평균적으로 400자의 길이를 가졌다.



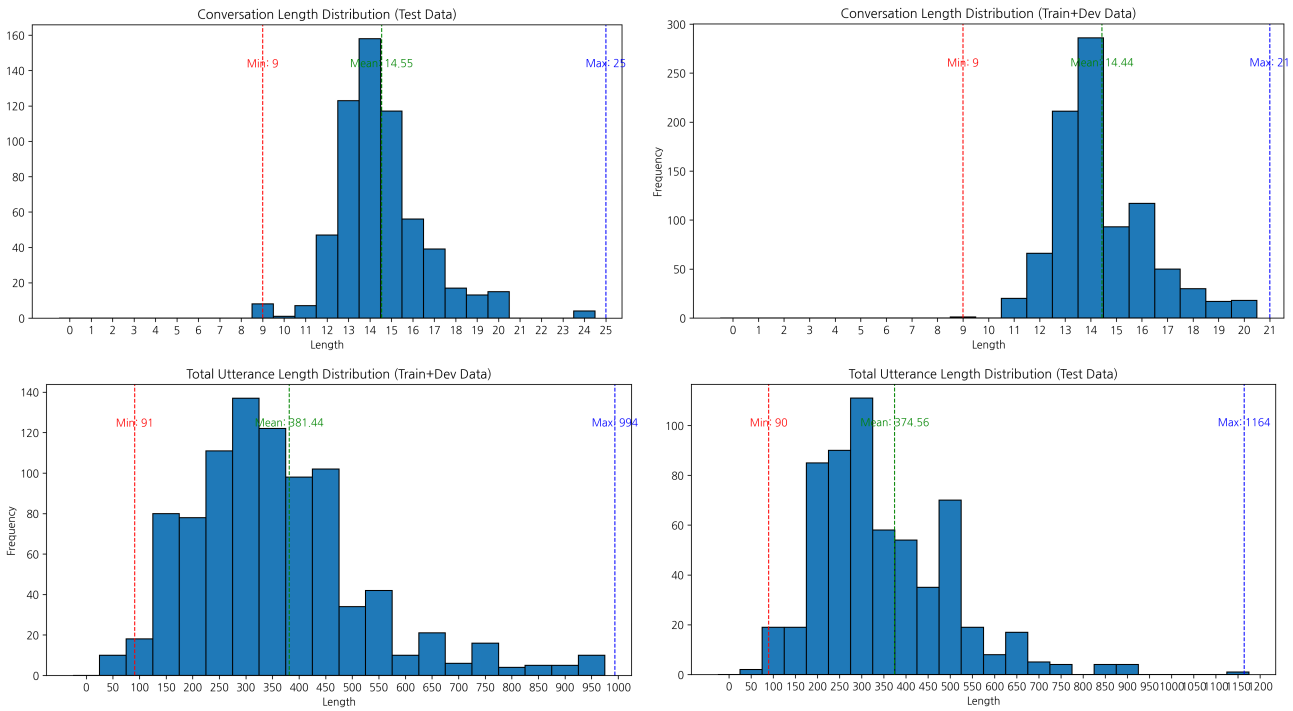
<그림 6> 원본 데이터셋의 대화 턴 수 분포
빨강색 점선은 최소 턴 수, 파랑색 점선은 최대 턴 수, 초록색 점선은 분포의 평균 턴 수를 나타냄
(좌: 훈련 데이터와 검증 데이터를 한 번에 분석한 결과, 우: 시험 데이터를 분석한 결과)



<그림 7> 원본 데이터셋의 전체 대화 길이 분포
빨강색 점선은 최소 길이, 파랑색 점선은 최대 길이, 초록색 점선은 분포의 평균 길이를 나타냄
(좌: 훈련 데이터와 검증 데이터를 한 번에 분석한 결과, 우: 시험 데이터를 분석한 결과)

② 전처리 된 데이터셋의 분포 (정제 및 전처리 이후)

<그림 8>은 정제 및 전처리 과정을 거친 데이터셋의 대화 턴 수와 전체 대화 길이 분포를 시각화한 것이다. 전처리 과정을 거치면서 **대화의 턴 수는 평균 5.5턴이 감소**하였으며, 원본 데이터셋의 분포와 달라진 것을 확인하였다. **전체 대화의 평균 길이는 약 20자 감소**하였으며, 원본 데이터셋의 분포가 유지되었다. <그림 8>을 통해 데이터 정제 및 전처리 과정을 통해 **대화의 턴 수와 길이가 감소하면서 적은 길이의 입력만으로도 모델을 효율적으로 훈련됨**을 확인할 수 있었다.



<그림 8> 데이터 정제 및 전처리 이후 데이터셋의 분포
(상: 대화 턴 수 분포, 하: 전체 대화 길이 분포,
좌: 훈련 데이터와 검증 데이터를 한 번에 분석한 결과, 우: 시험 데이터를 분석한 결과)

4. 모델 개요

1) 모델 선택 이유 및 전략

기존의 자연어 이해 및 생성 모델들은 대화 맥락을 이해하고 추론하는 데 어려움이 있다. 이에 대응하기 위해 본 참가팀은 **사전학습된 대형 언어 모델(LLM)을 미세 조정**하는 전략을 채택하였다. 사용된 모델들은 4종류로 [2]beomi/Solar-Ko-Recovery-11B, [3]x2bee/POLAR-14B-v0.2, [4]x2bee/POLAR-14B-v0.5, [5]chihoonlee10/T3Q-ko-solar-dpo-v7.0 이다. 이들 모델은 공통적으로 [6]Open Ko-LLM Leaderboard [7]에서 우수한 성과를 기록한 모델들로, 특히 **대규모 한국어 데이터에서 탁월한 이해도**를 보였다. 또한, 대회 규정에 맞게 **24GB 안팎의 메모리(GPU)로 실행 가능한 모델**들을 우선 선택하여 **효율성을 극대화**하였다.

2) 모델 아키텍처

① 기본 구조

사용된 4종류의 모델은 공통적으로 SOLAR 10.7B [8] 모델을 기반으로 추가적인 훈련을 통해 **한국어 능력이 강화**된 모델들이다. SOLAR 10.7B는 32개의 레이어로 구성된 Llama 2 [9] 아키텍처를 기본 모델로 사용하며, Mistral 7B [10]의 사전 훈련된 가중치로 초기화되어 훈련된 모델이다. SOLAR 10.7B는 깊이 업스케일링 (Depth Up-Scaling, DUS) 기술을 통해 기본 32개 레이어에서 시작하여 총 74개 레이어로 모델을 확장하였다. 추가된 레이어들은 기존 레이어의 가중치를 복사하여 초기화되며, 확장 후, 8개의 레이어를 전략적으로 제거하여 해당 모델은 최종적으로 66개의 레이어를 가진다.

② Parameter Efficient Fine-Tuning (PEFT) 전략

대규모 언어 모델(LLM)을 미세 조정하는 과정은 방대한 연산 자원과 메모리 용량이 요구된다. 이러한 점을 보완하기 위해 Parameter Efficient Fine-Tuning (PEFT) 기법을 적용하였으며, 이를 통해 **모델의 파라미터를 효율적으로 활용**하는 동시에, PEFT 기법을 사용하지 않았을 때와 비교하여 더 큰 파라미터를 가지는 대규모 언어 모델을 사용함으로써 **성능을 극대화** 할 수 있었다. 다양한 PEFT 기법들 중에서도 특히 LoRA (Low-Rank Adaptation) [2]와 QLoRA (Quantized Low-Rank Adaptation) [3]를 사용하여 **LLM의 일부 파라미터만 업데이트**하는 전략을 채택했다.

LoRA는 선형 계층을 저차원(low-rank) 근사치로 대체하여, 모델의 주요 파라미터 업데이트 없이도 미세 조정을 수행할 수 있도록 한 방법이다. 즉, 전체 모델의 파라미터가 아니라 모델 내 선형 변환에 사용되는 가중치의 일부분에서만 미세 조정이 이루어진다. 이는 **파라미터 수를 줄이면서도 높은 표현력을 유지**하며, **모델을 경량화할 수 있는 장점**이 있다.

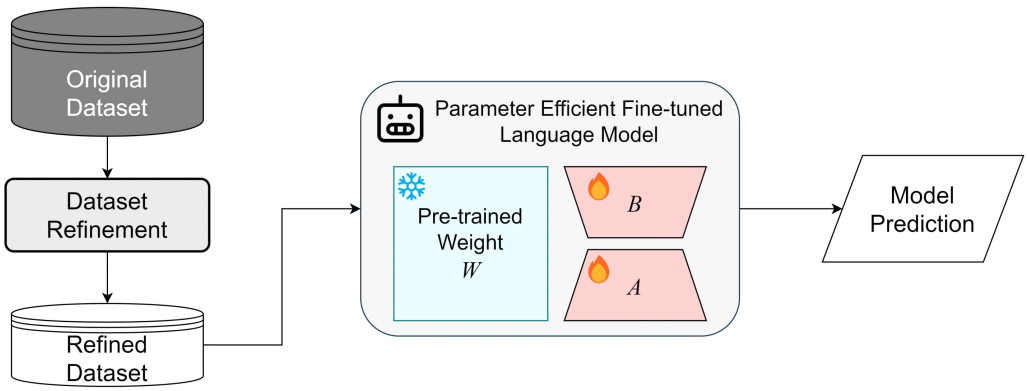
---

2) https://huggingface.co/beomi/Solar-Ko-Recovery-11B
3) https://huggingface.co/x2bee/POLAR-14B-v0.2
4) https://huggingface.co/x2bee/POLAR-14B-v0.5
5) https://huggingface.co/chihoonlee10/T3Q-ko-solar-dpo-v7.0
6) https://huggingface.co/spaces/choco9966/open-ko-llm-leaderboard-old

QLoRA는 LoRA 기법을 더욱 확장하여, NF4 방식의 양자화(Quantization)를 결합시킨 방법이다. QLoRA는 메모리와 연산 리소스를 더욱 효율적으로 사용하도록 도와주며, 양자화를 통해 제한된 자원에서 더 큰 파라미터를 갖는 LLM을 사용할 수 있으므로, 대화 맥락 추론과 같은 고차원적인 자연어 처리 작업에서 큰 장점을 갖는다.

<그림 9>는 PEFT 과정의 주요 단계들을 시각적으로 나타낸 것이다. Pre-trained Weight (W)는 사전학습된 LLM의 일부분으로, 해당 파라미터는 미세 조정 중에도 동결된 상태로 유지된다. A 및 B는 미세조정 과정에서 가장 중요한 두 부분으로, LoRA 및 QLoRA 기법을 통해 미세조정된 가중치를 나타낸다. A와 B는 낮은 랭크(차원)의 변환을 통해 **모델의 일반적인 성능을 해치지 않고, 특정 과제에 맞게 더 섬세한 조정**을 가능하게 한다. 이는 PEFT의 핵심 원리로, **전체 모델을 재학습하거나 거대한 자원을 낭비하지 않고도 우수한 성능을 이끌어낼 수 있다.**

본 참가팀은 4종류의 모델들을 기반으로 **다층적 학습 전략을 구성하여 11개의 모델을 개발**하였다. 사용한 전략은 <표 3>에 나타내었고, 학습 시 설정한 하이퍼 파라미터 변수 값은 <표 4>에 나타내었다.



<그림 9> Parameter Efficient Fine-Tuning (PEFT) 기반 모델의 학습 및 추론 과정

<표 3> 사용 모델 별 학습 전략

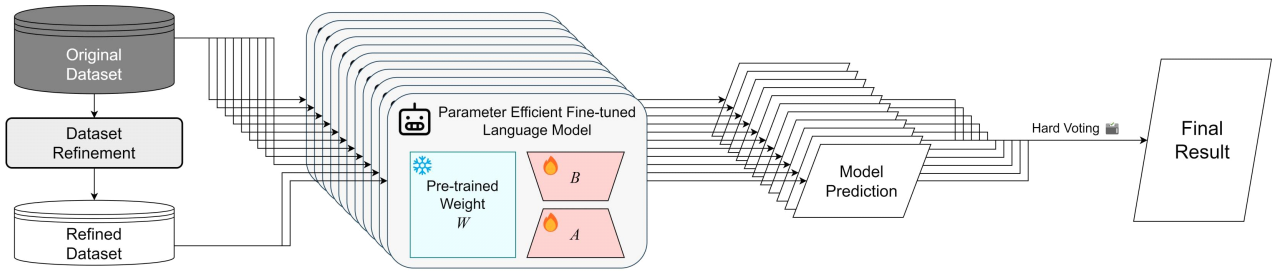
모델	PEFT 방법	학습률	데이터 정제	프롬프트 템플릿
x2bee/POLAR-14B-v0.2	QLoRA	2e-4	X	CustomRefOfficialTermDataset
x2bee/POLAR-14B-v0.5	QLoRA	2e-4	O	SystemRefOfficialTermDataset
x2bee/POLAR-14B-v0.5	QLoRA	2e-4	X	CustomRefDefinitionDataset
x2bee/POLAR-14B-v0.5	QLoRA	1e-4	X	CustomRefInstructionDataset
beomi/Solar-Ko-Recovery-11B	LoRA	2e-4	O	CustomRefDataset
beomi/Solar-Ko-Recovery-11B	LoRA	2e-4	X	CustomRefDataset
beomi/Solar-Ko-Recovery-11B	LoRA	1e-4	X	CustomRefDataset
beomi/Solar-Ko-Recovery-11B	LoRA	1e-4	X	CustomRefInstructionDataset
beomi/Solar-Ko-Recovery-11B	LoRA	1e-4	X	CustomRefDefinitionDataset
beomi/Solar-Ko-Recovery-11B	LoRA	1e-4	X	SystemRefOfficialTermDataset
chihoonlee10/T3Q-ko-solar-dpo-v7.0	LoRA	2e-4	X	CustomRefInstructionDataset

<표 4> 모델 학습 시 공통으로 사용된 하이퍼 파라미터 변수와 그 값

하이퍼 파라미터 변수		값
torch_dtype		bfloat16
seed		42
SFTConfig	epoch	4
	per_device_train_batch_size	1
	per_device_eval_batch_size	1
	weight_decay	0.1
	lr_scheduler_type	“cosine”
	warmup_steps	20
	neftune_noise_alpha	None
	gradient_accumulation_steps	64
	gradient_checkpointing	True
	gradient_checkpointing_kwargs	{"use_reentrant": False}
	max_seq_length	1024
LoraConfig	r	16
	lora_alpha	32
	lora_dropout	0.01
	target_modules	["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj", "lm_head"]

③ 앙상블 (Ensemble) 전략

성능의 일반화를 위해, 11개의 미세 조정된 모델들의 결과를 직접 투표(Hard Voting) 방식으로 앙상블 조합하였다. 직접 투표 방식은 입력으로 주어지는 대화와 질문에 대한 추론문의 정답 예측이 각 모델마다 다를 경우, 다수결에 따라 최종 추론문을 결정하는 방식이다. 이러한 앙상블 접근법은 **모델별 개별 약점을 보완**할 수 있고, 최종으로 예측하는 추론문이 일반성(generalization)을 가져 **리더보드 1위 달성**을 가능하게 하였다. <그림 10>은 여러 모델이 데이터를 입력받아 직접 투표 방식으로 최종 결과를 도출할 때까지의 과정을 나타낸다.



<그림 10> 직접 투표 (Hard Voting) 방식의 앙상블을 통해 최종 결과를 도출하는 과정

5. 평가 결과

본 모델의 최종 성능은 **98.0165%의 정확도**로 **리더보드 1위를 기록**하였으며, 이는 개발한 모델이 여러 차례의 실험과 앙상블을 통해 일반화되어 과제의 요구사항을 충실히 수행하였음을 의미한다. 본 모델의 성능은 데이터 정제 및 전처리 전략, 하이퍼파라미터 설정, 프롬프팅 전략, 그리고 모델 선택 전략에 큰 영향을 받았으며, 최신 연구 동향을 반영한 최적의 방법론을 적용함으로써 우수한 결과를 도출하였다.

1) 모델 개선 방안 및 추가 연구

현재 모델이 달성한 성능은 우수하지만, 본 참가팀은 일부 한계점 및 개선 가능성을 발견하였다. 본 절에서는 향후 추가 연구 및 개선 방안에 대한 제안을 서술한다.

① 어려운 추론 유형에 대한 성능 향상

본 참가팀은 학습된 모델이 일부 추론 유형에서 상대적으로 낮은 성능을 기록한 점을 확인하였다. 특히 '내적 동기'와 같은 복잡한 추론 유형의 경우, 정확도가 다른 유형에 비해 다소 낮았다. 이를 해결하기 위한 방법으로, 해당 유형에 대해 특화된 추가 데이터셋을 구축하거나, 더욱 정교한 프롬프팅 및 미세 조정 전략을 연구할 필요가 있다.

② 실시간 응용 가능성 검토

대규모 사전학습 모델(LLM)을 사용하는 현 모델 구조는 정확도 측면에서 매우 뛰어나지만, 실시간 응용에서 고려해야 할 지연 시간(latency) 이슈가 발생할 수 있다. 이를 해결하기 위해, 현 구조보다 더욱 경량화된 모델을 사용하거나 효과적인 지연 시간 최적화 기법을 적용해야 할 것이다.

③ 프롬프트 다양화 전략

현 모델은 추론문 유형에 상관 없이, 모두 같은 프롬프트 템플릿을 적용하였다. 향후에는 각 유형의 추론문에 대해 더욱 다양한 프롬프트 구성 방법을 연구하고 적용할 예정이다. 특히, 추론문 후보를 재작성하여 모호함을 없애고 명확성을 높이거나, 실제 대화 환경에 더 근접한 프롬프트 설계가 필요하다.

2) 실제 응용 분야로의 확장

맥락 이해 능력이 강화된 본 모델은 실제 응용 분야에서 다양하게 활용될 수 있는 잠재력을 가지고 있다. 다음은 모델이 성공적으로 적용될 수 있는 구체적인 분야와 시나리오를 검증된 추론문 유형 별로 서술한 것이다.

① [원인 측면] AICC (AI Contact Center) 플랫폼 강화

대규모 고객 콜센터에서 모델을 활용하여 고객의 발언을 이해하고, 맥락에 맞는 적절한 응답을 생성하

거나 정보를 추천함으로써 고객 만족도를 향상시킬 수 있다. 예를 들어, 고객이 “배송이 너무 지연되고 있어서 불만이 많아요”라고 말했을 때, 모델은 고객의 불만의 **원인**(배송 지연)을 이해하고, 이후에 발생할 수 있는 사건(환불 요청 등)을 예측하여 관련 정보를 즉시 제공하거나 해결 방안을 추천할 수 있다.

② [내적 동기] **스마트 홈 어시스턴트 강화**

스마트 홈 시스템에서 음성 명령 기반의 AI 어시스턴트는 사용자의 의도를 정확하게 이해하고 다양한 가전제품을 제어하는 역할을 한다. 예를 들어, 사용자가 “오늘 좀 더 따뜻하게 해줘”라고 말했을 때, 모델이 사용자의 **내적 동기**(추운 날씨로 인해 따뜻한 환경을 원함)를 이해하여 온도 조절기를 자동으로 조정할 수 있다.

③ [전제 조건] **게임 내 퀘스트 및 이벤트 트리거 시스템**

컴퓨터와 사용자가 텍스트 형태로 대화를 나누는 게임 내에서 대화의 맥락을 바탕으로 특정 이벤트나 퀘스트를 시작하기 전에 충족해야 하는 **전제 조건**을 파악하여, 보다 유연하고 동적이며 플레이어 맞춤형 경험을 제공할 수 있다. 이로 인해 플레이어가 다양한 경로로 목표를 달성하게 되고, 게임의 반복 플레이성을 높일 수 있다.

④ [감정 반응] **심리 상담 및 멘탈 헬스케어**

본 모델은 환자와의 심리 상담 챗봇으로 활용되어, 대화 맥락을 통해 환자의 발언에서 도출된 **감정 반응**을 파악하고 상담사가 목적에 맞는 맞춤형 상담을 제공하도록 도울 수 있다. 예를 들어서, 환자가 “최근 너무 힘들어서 무기력해요”라고 말했을 때, 모델은 ‘환자가 불안감이나 우울감을 느끼고 있다’는 **감정 반응**을 파악하고, 그에 맞는 상담 전략(예: 인지 행동 치료 추천, 명상 기술 적용 등)을 제안할 수 있다.

6. 모델 사용설명서

본 장은 모델을 학습하고 추론하기 위해 필요한 환경 설정 방법과 코드 실행 방법을 안내한다. 본 장의 내용은 모델 깃허브 리포지토리의 README.md 에서도 확인할 수 있다.

0) Requirements

코드 실행을 위해 아래와 같은 환경이 필요하다.

- Ubuntu 20.04.5 LTS
- Python 3.9.19
- Miniconda 24.1.2
- git

anaconda가 설치되어 있지 않은 경우에는 다음 명령어들을 참고하여 설치하여야 한다.

```
$ cd ~# 설치 파일을 다운로드할 경로로 이동 (to home directory)
$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh # Miniconda 설치 파일 다운로드
$ bash Miniconda3-latest-Linux-x86_64.sh # 설치 파일 실행
$ export PATH=~/miniconda3/bin:$PATH # 환경 변수 설정
$ source ~/.bashrc # Anaconda 설치 후 bash shell 환경 재설정
$ conda init # conda 초기화
$ conda --version # conda 버전 확인
```

1) 환경 설정

각각의 명령어들을 순차적으로 실행해야 한다.

① 개발 환경 설정

```
$ git clone https://github.com/oneonlee/KR-Conversation-Inference_Refined.git
$ cd KR-Conversation-Inference_Refined
$ conda create -n KR-Conversation-Inference python=3.9.19
$ conda activate KR-Conversation-Inference
$ pip install -r requirements.txt
```

② 한글 폰트 설치 (EDA 시 필요)

```
$ curl -o nanumfont.zip http://cdn.naver.com/naver/NanumFont/fontfiles/NanumFont_TTF_ALL.zip
$ sudo unzip -d /usr/share/fonts/nanum nanumfont.zip
$ sudo fc-cache -f -v
$ fc-list | grep Nanum
$ rm ~/.cache/matplotlib/fontlist*
```

2) 데이터셋 준비

먼저, 인공지능(AI)말평 과제 페이지에서 대화 맥락 추론 말뭉치(대화맥락추론_데이터.zip)를 다운로드 받아 **압축을 해제하지 않고** resource/data 디렉토리에 위치시켜야 한다. 그 다음 아래 명령어들을 순차적으로 실행해야 한다.

```
# 데이터셋 압축 해제
$ cd resource/data
$ unzip 대화맥락추론_데이터.zip
$ mv 대화맥락추론_데이터/대화맥락추론_train.json train.json
$ mv 대화맥락추론_데이터/대화맥락추론_dev.json dev.json
$ mv 대화맥락추론_데이터/대화맥락추론_test.json test.json
$ rm -r 대화맥락추론_데이터
```

```
# train.json과 dev.json을 합쳐 train+dev.json 파일 생성
$ head -n -1  train.json > temp.json
$ truncate -s -2 temp.json
$ echo ",">> temp.json
$ tail -n +2 dev.json >> temp.json
$ mv temp.json train+dev.json
$ cd ../../
```

```
# 데이터셋 전처리 및 정제
$ python run/refine.py
```

3) EDA (Exploratory Data Analysis)

데이터셋을 분석하기 위해 아래 명령어를 실행한다.

```
$ python run/EDA.py
$ python run/EDA.py --refined
```

분석 결과는 resource/EDA와 resource/EDA/refined 디렉토리에 저장되며, 아래와 파일들이 생성된다.

- o category_distribution.png : 추론문 유형별 분포
- o conversation_length_distribution.png : 대화 턴 수 분포
- o output_distribution.png : 정답 추론문 유형의 분포
- o total_utterance_length_distribution.png : 대화 전체 길이 분포
- o total_utterance_lengths.csv : 대화 별 전체 길이 데이터
- o wordcloud.png : 대화 내용 워드클라우드

4) 모델 학습 (Train)

학습에는 A100-PCIe-40GB GPU 1대가 사용되었다. 모델을 학습하려면 아래와 같이 실행한다.

```
$ sh scripts/train.sh
```

5) 모델 추론 및 앙상블 (Inference & Ensemble)

추론에는 RTX 4090 (24GB) GPU 1대가 사용되었다. 모델을 추론하고 최종 결과를 얻으려면 아래 명령어를 실행한다.

```
$ sh scripts/train.sh
```

최종 결과는 resource/results/predictions 디렉토리의 final-result.json 파일에 저장되며, 해당 파일 제출 시 리더보드에 성적이 반영된다.

참 고 문 헌

[1] Ding, Ning, et al. “Parameter-efficient fine-tuning of large-scale pre-trained language models.” *Nature Machine Intelligence* 5.3 (2023): 220-235.

[2] Hu, Edward J., et al. “LoRA: Low-Rank Adaptation of Large Language Models.” In *The Tenth International Conference on Learning Representations*, ICLR. 2021.

[3] Dettmers, Tim, et al. “QLoRA: Efficient Finetuning of Quantized LLMs.” *Advances in Neural Information Processing Systems* 36 (2024).

[4] Zamfirescu-Pereira, J. D., et al. “Why Johnny Can’t Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts.” *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023.

[5] Yang, Rui, et al. “GPT4Tools: Teaching Large Language Model to Use Tools via Self-instruction.” *Advances in Neural Information Processing Systems* 36 (2024).

[6] Fagbohun, Oluwole, Rachel M. Harrison, and Anton Dereventsov. “An Empirical Categorization of Prompting Techniques for Large Language Models: A Practitioner’s Guide.” *arXiv preprint arXiv:2402.14837* (2024).

[7] Park, Chanjun, et al. “Open Ko-LLM Leaderboard: Evaluating Large Language Models in Korean with Ko-H5 Benchmark.” *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2024.

[8] Kim, Dahyun, et al. “SOLAR 10.7B: Scaling Large Language Models with Simple yet Effective Depth Up-Scaling.” *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*. 2024.

[9] Touvron, Hugo, et al. “Llama 2: Open foundation and fine-tuned chat models.” *arXiv preprint arXiv:2307.09288* (2023).

[10] Jiang, Albert Q., et al. “Mistral 7B.” *arXiv preprint arXiv:2310.06825* (2023).