Final Project (CALL-E)

8조 MIML

[1] 프로젝트 주제

메신저 단체 대화방에 적절한 이미지를 생성하기 위해 대화를 요약하고, 요약한 대화문을 활용해 이미지를 생성하는 프로젝트

[2] 프로젝트 팀 구성 및 역할

조원	역할
임동진	대화 요약 feasibility check, 모듈화 및 metric 구현, backend & product serving, github repository 관리
정재윤	채팅 데이터 전처리, Frontend
조설아	Text-to-image 모델 모듈화 및 metric 구현, image dataloader 구현, inpur format 변경 실험
허치영	대화 요약 모델 구현, documentation, model search
이보림	Text-to-image fine-tuning, image crawling 구현, 이미지 데이터셋 제작, docker

[3] 프로젝트 수행 절차 및 방법

1. 데이터셋

1) AI Hub 한국어 대화 요약 데이터셋

- Train data: 279,992 / Validation data: 35,004
- 사용 목적: 대화 요약 모델 fine-tuning


```
"body": {
    "dialogue": [
        {
            "utteranceID": "U1",
            "turnID": "T1",
            "participantID": "P1",
            "date": "2020-07-23",
            "time": "10:52:02",
            "utterance": "내일이 #@이름# 엄마 생일임.",
        },
        {
            "utteranceID": "U2",
            "turnID": "T1",
            "participantID": "P1",
            "date": "2020-07-23",
            "time": "10:52:05",
            "utterance": "저녁에 외식할까 함."
        },
        {
            "utteranceID": "U3",
            "turnID": "t1",
            "participantID": "P1",
            "date": "2020-07-23",
            "time": "10:52:06",
            "utterance": "시간 ㅇㅋ?"
        },
        {
            "utteranceID": "U4",
            "turnID": "T2",
            "participantID": "P2",
            "date": "2020-07-23",
            "time": "10:53:08",
            "utterance": "강의만 제때 끝나면 저녁에 시간돼요",
        },
        {
            "utteranceID": "U5",
            "turnID": "T3",
            "participantID": "P3",
            "date": "2020-07-23",
            "time": "11:15:20",
            "utterance": "저도 가능해요",
        }
    ]
    "summary": "엄마 생일을 맞아 모두 모여 외식할 수 있는 저녁 식사 시간을 정하고 있다."
},
...
]
}
```


2) KorSTS

- Train data: 5,749 / Validation data: 1,500
- 사용 목적: 대화 요약 모델 성능 평가 지표에 사용할 sentence BERT 학습

Example	English Translation	Label
한 남자가 음식을 먹고 있다. 한 남자가 뭔가를 먹고 있다.	A man is eating food. A man is eating something.	4.2
한 비행기가 착륙하고 있다. 애니메이션화된 비행기 하나가 착륙하고 있다.	A plane is landing. A animated airplane is landing.	2.8
한 여성이 고기를 요리하고 있다. 한 남자가 말하고 있다.	A woman is cooking meat. A man is speaking.	0.0

3) Crawled Image

- Train data: 15,373 / Validation data: 1,708
- 사용 목적: Text-to-image 모델 fine-tuning

Image

Text

Actor Kwak Do-won appeared Live Nahonsan Mountain said live Jeju Island second rest conveying various activity saw TV

4) Google Conceptual Caption
- Validation data: 15,840
- 사용 목적: Text-to-image 성능 검증

where 's the best place to show off your nails ?
right in front of the castle , of course !

hat combines elements of a simple vegetable and dish

5) 카카오톡 단체 채팅방 크롤링 데이터
- 사용 목적: CALL-E 전체 모델 성능 평가

[초코] [오후 5:32] 난 가능한데
[샐리] [오후 5:32] 하...
[샐리] [오후 5:32] 그때 일정 봐야할거같은데
[문] [오후 5:41] 나 원주감 ㅎㅎ 나중에봐
[문] [오후 5:42] 담주에 다들 괜찮음??
[초코] [오후 5:42] 다음주 주말은 너무 좋지
[초코] [오후 5:43] 15일 이후부턴 괜찮음
[샐리] [오후 6:22] 나
[샐리] [오후 6:22] 17일날 끝남
[샐리] [오후 6:22] 개꿀
[샐리] [오후 6:41] 굿
[브라운] [오후 7:36] ㅇㅋ
[초코] [오후 7:39] 그럼 18일로 알고있을게요
[문] [오후 8:38] 나 18일 약속있음
[샐리] [오후 8:44] 19일 ㄱ

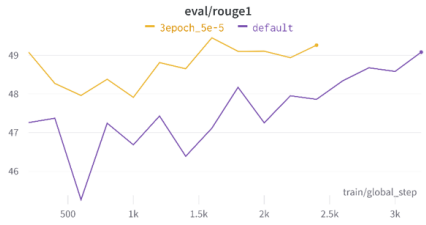
2. 대화 요약 Model

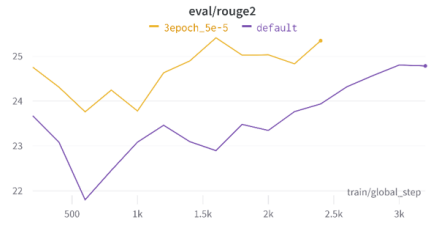
1) Model Search

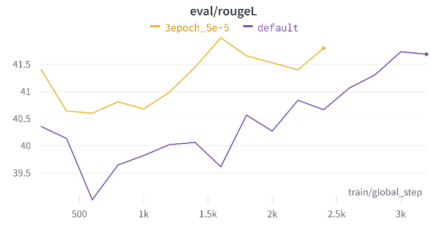
- Abstractive Summarization: 대화 요약의 특성 상 중요한 부분을 추출하는 방식은 제대로 수행되지 못할 것으로 예상하여 생성 요약 방식을 채택했습니다.
- KoBART: KE-T5 에 비해 상대적으로 작은 모델 크기로 좋은 성능 나타냈기에 KoBART 를 채택했습니다.

2) Feasibility check

- 모델: BART
- 학습 데이터셋: SAMSum
- 목적: 대화 요약 가능성 확인







〈 BART Performance on SAMSum 〉

Model	ROUGE-1	ROUGE-2	ROUGE-L
BART-large-SAMSum	53.434	28.744	44.185

〈 BART Performance in paperswithcode.com 〉

비교 결과, paperswithcode 의 BART large 와 매우 유사한 결과를 나타냈고, BART 의 대화 요약 수행이 가능함을 확인했습니다.

3) Metric 선정

- ROUGE

  Summarization 에서 가장 널리 활용되는 평가 지표

  사용하지 않은 이유: 한국어 접사, 동의어를 고려하지 않아 정확한 성능 평가 불가

- RDASS

  문서, 예측 요약, 정답 요약 벡터 사이의 cosine similarity 를 이용하여 성능 평가

  사용하지 않은 이유: 대화 요약 특성 상 문서(대화기록)와 요약 문장 사이의 cosine similarity 정확하게 평가 불가

- **Cosine similarity**

  정답 요약과 예측 요약 사이의 cosine similarity 를 이용하여 성능 평가

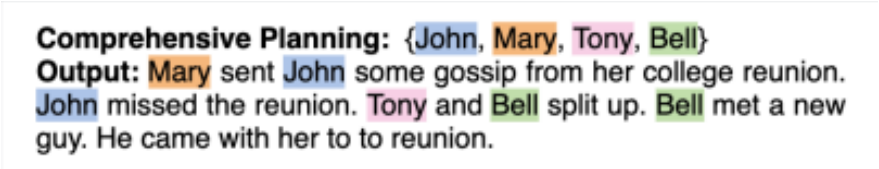
  데이터셋 평균 cosine similarity 와 임계 값(0.6) 이상의 개수로 정량 평가

  0.6: KorSTS 기준 score 3.0(cosine similarity 0.6) 이상인 경우 유사한 문장으로 판단되어 기준 값으로 설정

  사용 이유: RDASS 에서 문제되었던 문서와 요약문 사이의 cosine similarity 를 제거하여 정답 요약과 예측 요약에만 집중하도록 했습니다.

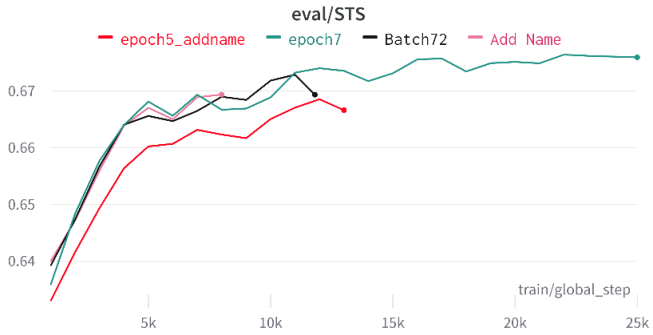
4) 성능 향상 실험

- Named Entity Planning



〈 *Named Entity Planning Example* 〉

논문(Controllable Neural Dialogue Summarization with Personal Named Entity Planning)에 따라 모델의 입력에 대화 참가자 ID 를 추가하는 방식을 실험했습니다. 실험 결과 약 0.1 가량의 평균 cosine similarity 가 하락하여 사용하지 않았습니다.
(Add Name: Named Entity Planning, Batch72: baseline)



〈 *Average of Cosine Similarity* 〉

- model.generate()

학습이 완료된 모델을 이용해 요약을 생성할 때 요약의 품질을 향상시키기 위해 요약 생성 시 몇 가지 옵션을 수정했습니다.

generate() arguments	Avg. Cosine Similarity	# of Cosine Similarity >0.6
Baseline	0.6443	23,121
temperature(0.7)	0.6453	23,438
no_repeat_ngram_size(3)	0.6482	23,238
temperature(0.7) + no_repeat_ngram_size(3)	**0.6487**	**23,477**

```
outputs = model.generate(input_ids,
                         num_beams=5,
                         max_length=64,
                         attention_mask=attention_mask,
                         top_k=50,
                         top_p=0.95,
                         no_repeat_ngram_size=3,
                         temperature=0.7
                         )
```

〈 *최종 선택 option* 〉

• Num_beams: beam size 를 의미, 1 이상으로 설정 시 beam search 수행 (greedy decoding 보다 좋은 결과 얻기 위해서 사용
• Max_length: 생성되는 문장의 최대 길이 (정답 요약의 최대 길이가 64 를 넘지 않아 비슷한 수준의 길이로 설정)

• Top_k: token 들 중 예측될 확률이 상위 50 개인 것들에 대해서만 redistribution
• Top_p: token 들 중 예측될 확률이 0.95 이상인 것들에 대해서만 sampling
• No_repeat_ngram_size: 해당 n gram 반복되지 않도록 설정 (무의미한 단어 반복 막기위해 3 으로 설정)
• Temperature: 모델이 생성하는 문장의 무작위성을 약하게 하기 위해서 0.7 로 설정

3. Text-to-Image 모델

1) model search

- LAFITE: 상대적으로 작은 모델 사이즈 (parameter: 7500 만개)로 서비스화에 적합. Language Free 모델로 이미지만으로 학습이 가능하며 여러 pretrained model 들을 제공하기 Fine-tuning 하기에 용이
- Latent Diffusion Model: 4 억 개의 image-text 쌍으로 이뤄진 거대한 데이터셋을 학습한 모델. GAN 을 뛰어넘는 높은 image 합성 성능
- minDALL-E: DALL-E 에서 착안한 모델. 높은 zero-shot 성능. DALL-E 와 달리 적은 리소스로 학습 가능

모델 선정 기준

• CLIP Score (높을수록 좋은 성능)
텍스트-이미지 사이의 상관 관계 평가

• FID Score (낮을수록 좋은 성능)
생성된 이미지와 정답 이미지의 유사도 평가
FID, Inception Score 와 달리 요약 문장을 활용 가능

	minDALL-E	Lafite	Latent Diffusion(LDM)
CLIP score	0.3193	0.2722	0.2170
FID score	322.235	467.279	399.945

임의의 test dataset 100 개 이미지에 대해서 성능을 평가했으며, CLIP score 가 가장 높은 minDALL-E 로 최종 모델을 채택했습니다.

2) 성능 향상 실험

Input Sentence Transformation

- Papago API 를 활용하여, 한국어 요약 문장을 영어로 번역한 후 총 12 가지 방식으로 변형하여 실험을 진행했습니다.

	접두어	문장 전처리	문장 재구성
내용	① 접두어 없음	④ 원본 문장	⑦ 띄어쓰기로 구분
	② A painting of	⑤ 명사/동사/형용사 추출	⑧ 쉼표로 구분
	③ A painting =	⑥ 불용어 제거	⑨ +기호로 구분

〈 문장 변형 방식 〉

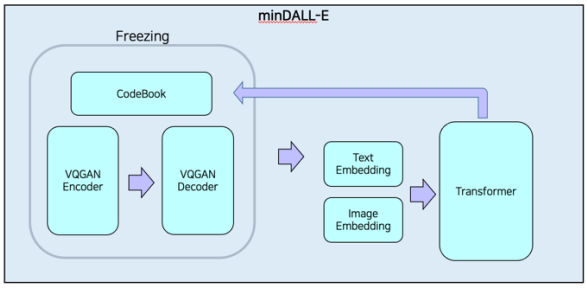
CLIP score 와 이미지 품질이 반드시 비례하지는 않아 CLIP score 가 0.2 인 이미지에 한해 human evaluation 을 실시했습니다.

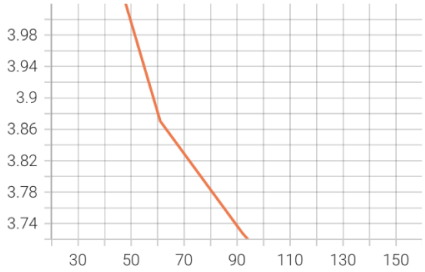
• BEST: 불용어 제거 + 쉼표로 문장 성분 구분

문장 변형 방법 조합	문장	CLIP score
① + ④ + ⑦	I went home right away today, but I couldn't rest well.	0.2002
① + ⑤ + ⑦	went home right away today n't rest well	0.2017
① + ⑤ + ⑧	went, home, right, away, today, n't, rest, well	0.2126
① + ⑥ + ⑦	went home right away today , could n't rest well .	0.1899
① + ⑥ + ⑧	went, home, right, away, today, ,, could, n't, rest, well, .	0.2100
② + ④ + ⑦	A painting of I went home right away today, but I couldn't rest well.	0.2475
② + ⑤ + ⑦	A painting of went home right away today n't rest well	0.2521
② + ⑤ + ⑧	A painting of went, home, right, away, today, n't, rest, well	0.2719
② + ⑥ + ⑦	A painting of went home right away today , could n't rest well .	0.2570
② + ⑥ + ⑧	A painting of went, home, right, away, today, ,, could, n't, rest, well, .	0.2727
③ + ⑤ + ⑨	A painting = went + home + right + away + today + n't + rest + well	0.2536
③ + ⑥ + ⑨	A painting = went + home + right + away + today + , + could + n't + rest + well + .	0.2521

〈 문장 변형 방식 별 성능 비교 〉

2) 모델 학습

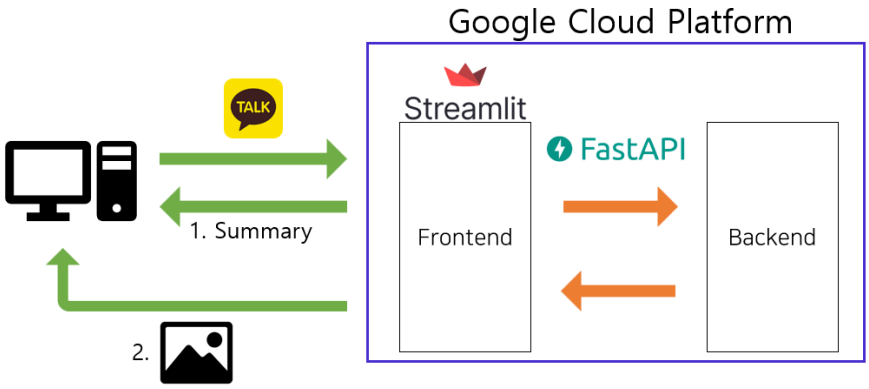




• Cross-entropy Loss 사용했고, Early Stopping과 Tesorboard를 이용하여 실험을 관리했습니다.
• 학습에 다양한 이미지가 사용되어 배치 크기를 작게 설정한 경우 성능 저하를 불러와 512로 설정하여 학습했습니다.
• 이미지에는 큰 변화가 없다고 판단하여 minDALL-E의 VQGAN 부분은 freezing한 후 학습을 진행했습니다.
• CLIP score와 실제 생성된 이미지의 품질이 항상 일치하지 않았습니다. 또한 FID score의 평가가 반영되지 않았으며 텍스트는 이미지를 묘사하는 내용이 아닌 대화 내용을 담고 있기에 CLIP score만으로는 가장 좋은 이미지를 판단할 수 없다고 생각하여 3장의 이미지를 생성하고 사용자가 그 중에서 선택하는 방식으로 이미지를 생성했습니다.

4. Product Serving

동작 과정: txt 파일 형식의 Dialogue 를 string 형식으로 변환 → Backend 에 요약문 생성 요청 → Respond 로 요약문 출력 →
Request 로 Backend 에 이미지 생성 요청 → 이미지 출력



Frontend (Streamlit)

- Streamlit: python 으로 쉽게 빌드 가능. 여러 component 를 이용해 간단하게 웹사이트 제작 가능

Backend (FAST API)

1) 2단계 서비스 제공

사용자의 체감 대기 시간을 줄이기 위해 중간 결과물 (대화 요약문)을 먼저 제공하고 최종 결과물을 보여주는 방식을 선택했습니다. Request를 2번 보내야 한다는 단점이 존재하지만, 사용자는 요약문을 보며 흥미를 느낄 수 있다고 판단하였고, 이는 체감 대기 시간을 줄일 수 있는 좋은 방법이라고 생각했습니다.

2) 모델 상시 Load

이미지 생성 모델을 Request 마다 불러오기에는 오랜 시간 기다려야 하기 때문에 backend 에서 미리 불러와 Request 가 올 때마다 미리 불러와져 있는 모델을 활용해 이미지를 생성할 수 있도록 구현했습니다.

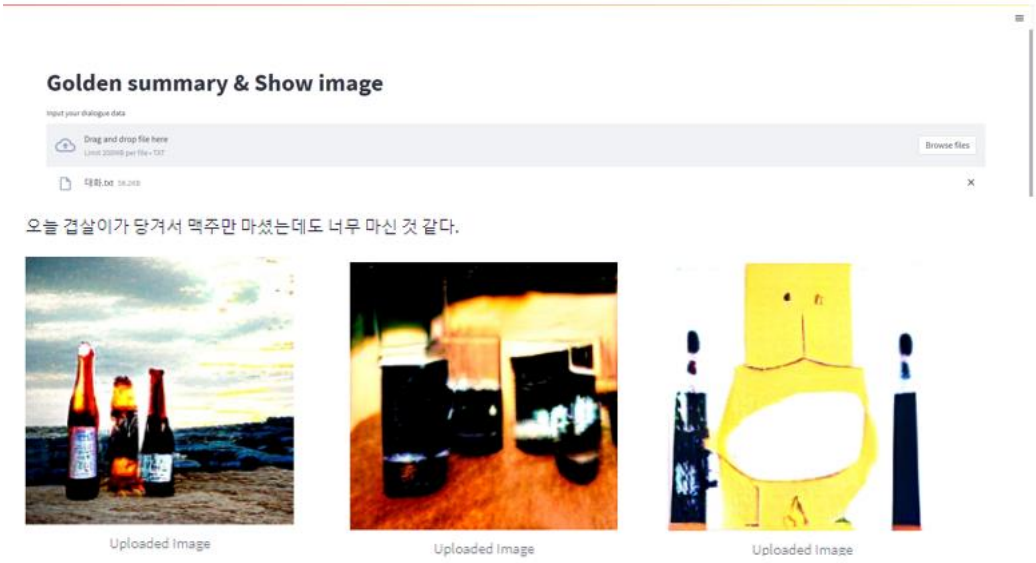
GCP

활용 이유: 무료 GPU 활용 가능 (CPU 에서 추론 시 서비스 시간 과도하게 길어지는 문제 발생).
무료로 충분히 긴 기간 동안 service 운영 가능

- Machine 유형 : n1-standard-4 - GPU : NVIDIA Tesla T4 1 개

[4] 프로젝트 수행 결과

Summary: “오늘 겹살이가 당겨서 맥주만 마셨는데도 너무 마신 것 같다.”



[5] 자체 평가 의견

- 잘했던 점

1. 프로젝트의 시작에 앞서 각 작업에 대해 계획을 세우고 인원을 알맞게 분배한 후 시작했습니다. 그 덕분에 일정에 맞게 프로젝트를 완수할 수 있었습니다.
2. 과제 난이도가 어려웠음에도 불구하고 생각했던 것보다 좋은 결과물을 만들어 냈습니다. 사람이 직접 대화 요약문을 이용해서 그림을 그리는 것도 매우 어려운 작업이었으나 요약문의 핵심을 잘 포착해내는 그림을 생성했습니다.
3. 혼자 진행하기 어려웠던 작업에 대해 다 함께 고민하고 해결해냈습니다. 각 팀원의 작업 중 어려웠던 부분들에 대해 함께 해결 방법을 생각해 해결하지 못하리라 생각했던 부분들도 잘 해결할 수 있었습니다.

- 아쉬웠던 점

1. 대화 요약 모델을 학습 시킬 때 선행 연구의 부족과 아이디어의 고갈로 인해 많은 기법들을 실험하지 못했으며, 더 다양한 모델에 대해 실험하지 못했던 것이 아쉬웠습니다.
2. 이미지 생성 모델을 huggingface hub 에 업로드하고자 시도했으나 성공하지 못했던 것이 아쉬웠습니다.
3. 계획 수립에 있어 모델 구현에 너무 많은 시간을 할애하여 serving 에 대해서 충분히 다양한 실험을 진행하지 못했던 것이 아쉬웠습니다.

개인회고: 임동진

나의 목표

배우지 않았던 모델을 Baseline Code 없이 처음부터 짜보며 실력을 검증하는 것이 첫 목표였습니다.
또한 이전에 배웠던 모든 경험 및 교훈을 적극적으로 활용해보는 것이었습니다.
마지막으로, 이전 Backend Project 때는 AI Model 과 서비스를 연결하지 못했었는데, 이번에는 만든 프로젝트를 실제 Backend 기술과 연동시켜 보고 싶었습니다.

무엇을 어떻게 했는가?

Jupyter Notebook 을 활용하여 Dialogue Summarization Task 에 대한 Feasibility Check 를 수행하였습니다.
이번 Project 가 실제로 구현할 수 있는 Task 라는 것이 판명된 이후, 허치영 팀원이 짜준 Baseline Code 를 모듈화하고 여러 가지 실험을 해보았습니다.
ROUGE 가 한국어에 잘 맞지 않는 Metric 이라는 생각이 들어 이번 Project 에 가장 적절할 것 같은 Metric 에 대해 알아보고, 직접 구현하여 활용하였습니다.
Hyperparameter Tuning 및 Input Data 형식을 변경시키는 등 최대한 모델의 Size 증가 없이 성능을 향상하기 위해 많은 실험을 수행하였습니다.
마지막으로 GCP 라는 새로운 Cloud Platform 을 활용하여 Product Serving 을 수행했는데, AWS 를 공부할 때와는 다르게 최대한 "어떤 방식으로" 동작이 이루어지는가를 이해하며 활용하였습니다.

아쉬운 점 및 다음 대회 때 시도할 점

1. <u>상용화할 모델이라면 모델 배포 이후 실험도 많이 해보자</u>

모델도 많은 실험을 수행할수록 좋은 모델을 선택할 수 있듯 Cloud 나 서버에 모델을 올리고 많은 실험을 해볼수록 서버 동작 상황을 빨리 파악할 수 있고, 안정성을 올릴 수 있을 것 같습니다.
실제로 이번 Project 에서 1 개 모델만을 상시 Load 하다 보니 여러 명의 User 가 동시에 Request 를 보낼 경우 먼저 Request 를 보낸 User 의 작업이 끝날 때까지 대기 해야 한다는 단점이 존재하였습니다.
배포 이후 실험을 많이 했다면 이런 부분을 빨리 파악하고 해결 방법을 찾아봤을 텐데 그러지 못했던 것이 아쉬웠습니다.

2. <u>제대로 된 협업 수행</u>

이번에 "대화 요약"팀 간의 협업은 잘 되었다고 생각합니다. 하지만 Pipeline 을 구축할 때 "대화 요약"팀과 "Text-to-Image" 팀과의 협업은 잘 안된 것 같다고 느꼈습니다.
지금까지 2 개의 Task 에 대해 아예 팀을 나눠 Pipeline 을 구축하는 경험은 없었다 보니 이런 부분에서 경험 부족이 드러났던 것 같습니다.
앞으로 대규모 프로젝트 같은 경우 우리 팀의 협업 방식만 신경 쓰는 것이 아니라 Pipeline 전체를 이해하고 이를 바탕으로 어떻게 다른 팀과 협업해야 하는지 생각하는 과정이 필요할 것 같습니다

3. <u>주제를 정할 때 인간이 Task 를 수행한다고 가정해보기</u>

"나는 3 박 3 일 여행을 가서 피곤하다"
이 문장을 사람에게 Image 로 그리라고 하면 큰 어려움을 느낄 것입니다.
AI 라는 것은 사람의 뇌를 따라가려고 만든 기술. 즉, 사람이 하기에도 어려운 일이라면 컴퓨터 또한 하기 어려운 일이라는 것을 느꼈습니다.
프로젝트를 시작하기 전에는 컴퓨터는 뛰어난 능력을 갖췄으니 상상치도 못한 결과를 낼 수 있을 것으로 생각하였지만,, 이것은 틀린 생각이었습니다.
앞으로는 프로젝트 주제를 정할 때 "사람이 시간을 오래 들이면 이 일을 수행할 수 있는가?"를 먼저 생각해보는 과정이 필요할 것 같습니다.

개인회고: 정재윤

- **이번 프로젝트에서 나의 목표는 무엇이었는가?**

5 개월간 인공지능에 대한 여러 개념들을 배웠고, 4 번의 team project 를 통해 다양한 Task 를 경험했습니다. 이를 바탕으로 실생활에서 사용가능한 AI 서비스르 제작하고 구현부터 배포까지 전체 흐름을 경험해보는 것을 목표로 삼았습니다.

- **나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?**

AIHub 의 채팅요약 데이터와 KorSTS 를 활용하여 약 28 만개의 채팅 데이터를 수집할 수 있었습니다. 하지만 실제로 입력되는 채팅들은 수집한 데이터보다 훨씬 큰 대규모의 채팅도 존재하고 채팅의 다양한 주제들을 커버하기 어렵다고 판단해 실제 오픈채팅방 데이터를 수집하여 보완하였습니다. 그리고 streamlit 을 활용하여 product serving 을 진행하였는데, 어떤 흐름으로 서비스가 동작하는지를 이해하며 수행하였습니다.

- **나는 어떤 방식으로 팀에 기여했는가?**

처음 수집한 채팅방 데이터는 모델에 입력하기에 적합하지 않은 구조이기에 전처리 과정을 반드시 거쳐야 했습니다. 모델에 입력하기 적합한 구조로 변환하고 개인정보들은 punctuation 을 추가해주는 등 전처리를 해주었습니다. 또한 최근 대화가 해당 채팅방을 가장 잘 대표할 것이라 판단하여 각 채팅데이터에서 최근시간대의 채팅 데이터만 사용하였고, 이는 중구난방의 채팅 데이터가 이상한 방향으로 요약되는 것을 방지해주었습니다.
추가로 streamlit 을 활용하여 웹페이지를 구현하였습니다.

- **마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

1. 채팅 데이터의 전처리 과정에서 개인정보가 완벽히 전처리 되지 못한 것이 아쉬웠습니다. 입력 받는 채팅 데이터의 형태 등 미처 전처리 되지 못한 채팅 데이터가 처리될 수 있는 방안을 고려해보았으면 하는 아쉬움이 남았습니다.

2. EDA 를 하지 않은 점이 아쉬웠습니다. 입력되는 데이터를 보고 이번 Task 에서 EDA 는 그닥 중요하지 않다고 판단하여 넘어갔습니다. 하지만 Hyperparameter Tuning 을 하면서 EDA 의 중요성을 다시금 깨달았습니다. 채팅의 길이와 개수 그리고 시간대 등 간단히라도 분석했다면 Hyperparameter Tuning 및 협업을 하는데 있어 효율적으로 일처리가 진행되었을 것이었고, 이미 EDA 의 중요성을 알고 있었기에 더욱 아쉬운점으로 남았습니다.

3. Streamlit 을 충분히 공부하지 못한채로 구현에 돌입
구현이 간단하다는 이유로 Streamlit 을 이용하여 Frontend 를 구현을 시작했습니다. 하지만 다운로드 기능을 구현할 때 원하는 Image 를 선택하고 다운로드 버튼을 누르기 전인데도 계속 페이지가 새로고침이 되는 이슈가 있었는데, 이는 버튼을 누르면 전체 Streamlit 페이지가 reload 되는 이유였습니다. 이로 인해 이전에 계획한 시간보다 더 오래 시간을 소비하였고 높은 완성도의 페이지를 구현해내지 못해 아쉬움이 남았습니다.

대회 후기

1. 내일 말고도 다른 팀원의 task 도 신경써주기
전체 task 에서 2 명은 채팅로그 요약, 2 명은 Text-to-Image, 그리고 저는 전체 데이터 수집 및 전처리를 담당하였습니다. 이번 Task 가 2 개의 세부 Task 로 나눠지다보니 어느때보다 협업의 중요성이 높았습니다. 제가 그 중간에서 Pipeline 전체를 이해하고 조율을 했었더라면 더 원활하게 프로젝트가 진행될 것 같은 아쉬움이 남았습니다. 내가 맡은 일을 잘 해해내는 것 중요하지만 전체 흐름을 파악하고 팀원이 더 나은 역량을 펼칠 수 있게 도와주는 것 또한 이번 프로젝트에서 더 잘 느낄 수 있었습니다.
2. 주제 선정시 인간 스스로도 할 수 있는지 생각해보기
지금 다시 생각해보면 기획의도는 좋은 프로젝트였습니다. 그러나 문장이 주어졌을 때, 모든 문장들을 인간이 완벽하게 한장의 그림으로 그려내기에는 무척 어려움을 느낄 것입니다. 인간이 할 수 있는 일을 기계에 학습을 시켜 서비스를 제공해야 하는데, 인간이 하기에도 어려운 일이면 컴퓨터도 어려운 일이라는 걸 느꼈습니다. 앞으로는 주제선정시에 이를 고려하여 신중히 생각해보는 과정이 선행되어야 함을 느꼈습니다.

개인회고: 조설아

- 이번 프로젝트에서 나의 목표는 무엇이었는가?

  지난 대회에서 **클린 코드 작성과 함수화 및 모듈화**에 미숙하여 아쉬웠기 때문에 이번 대회에서는 유의미한 변수명을 사용하고 기능별로 함수를 만들어서 utils.py 파일이나 적절한 파일 안에 비슷한 기능끼리 모아서 사용하는 것이 첫번째 목표였습니다. 그리고 모듈화할 때마다 잘 돌아가는지 **테스트**해서 프로세스가 꼬이는 일이 없도록 하는 것이 두번째 목표였습니다. 또한 KLUE 대회 때 모델링이 아닌 문장의 전처리를 통해 크게 성능을 향상 시킨 바가 있었기에 이번 프로젝트에서도 **문장의 전처리와 변형을 통해 성능을 개선**시키고자 한 것이 세 번째 목표였습니다. 마지막으로 전반적인 **AI 서비스 개발** 과정을 경험해보면서, 모델링에만 온전히 집중할 때와 달리 **어떻게 선택과 집중이 이뤄져야 하는지** 되돌아 보는 것이 네 번째 목표였습니다.

- 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

  그리고 데이터셋과 데이터로더를 구축한 후 이를 모듈화하여 기능들을 구조화하였습니다. 처음 하나씩 뜯어볼 때에는 주피터 노트북으로 잘 실행이 되는지 테스트를 진행한 후 **유의미한 변수 명을 사용하고 기능별 함수를 구현**하여 파이썬 파일로 정리했습니다. 또한 **주석을 다는 습관**을 들여 직접 설명하지 않고 코드만 보아도 누구나 알 수 있도록 하려고 노력했습니다.

  한편 제가 맡은 파트인 Text-to-Image 에서 사용한 모델은 minDALL-E 로, 영어로 학습된 모델이었기 때문에 문장의 전처리 및 변형은 한국어 문장을 영어로 번역한 후에 진행하였습니다. 저는 속도 측면에서 Google Translate 보다 빠른 papago API 를 사용했습니다. 그리고 다양한 조합의 변형 방법을 생각해 **접두사, 불용어 제거, pos 태깅, 문장 성분 재구성 등 다양한 방면에서 실험**을 진행했으며 **정량적 지표(CLIP Score)를 활용하여 객관성을 확보**하려고 했습니다.

- 나는 어떤 방식으로 모델을 개선했는가?

  저희 모델에 입력으로 들어오는 문장은 일반적인 Text-to-Image 태스크와 달리 **추상적인 내용의 문장**이 주를 이뤘습니다. 따라서 모델이 문장을 그대로 시각화했을 때 문장을 이해했다고 보기는 어려운 이미지들이 생성되었습니다. 그래서 저는 **모델이 알아들을 수 있게 문장을 재구성할 필요성**을 느꼈습니다.

  Text-Image 데이터셋의 텍스트가 대개 시각적 묘사이고 명사구이기 때문에 "A painting of"와 같은 prefix를 덧붙여 시각적인 묘사의 명사구로 바꾸어주는 방식을 시도해보았습니다. 그리고 모델이 맥락을 잘 파악하지 못하므로 강조하고 싶은 문장 성분(명사, 동사, 형용사)만을 뽑거나, 불용어를 제거하는 시도도 해보았습니다. 그렇게 추출된 문장을 쉼표로 잇거나, 띄어쓰기로 잇는 등의 재구성 방식 또한 취해보았습니다. 또, 수학 수식처럼 구성하여 "A painting = word1 + word2"와 같이 만들어보기도 했습니다.

  다양한 조합의 문장 각각에 대해 이미지를 생성하고 CLIP Score 를 산출한 결과, 0.2 미만의 경우에는 판단이 힘든 이미지가 생성되어 유의미하지 않다고 판단하였습니다. 그러나 0.2 이상의 점수를 낸 이미지의 경우에는 Score 와 Human Evaluation 이 비례하지 않았습니다. 그렇다고 반비례하지도 않고 그다지 규칙성이 없어 큰 문제였습니다.

  왜냐하면 CLIP Score 만을 판단 기준으로 내세우기에는 앞서 말했듯 추상적 묘사가 많아, CLIP Score 가 저희 프로젝트에 꼭 맞는 지표가 아니었기 때문입니다. 그러나 CLIP Score 보다 적절한 지표가 달리 존재하지도 않았기 때문에 저희는 CLIP Score 0.2 이상인 이미지 대상으로 조원끼리 Human Evaluation 을 진행했습니다. 문장 속 단어들을 최대한 많이 담은 그림을 좋은 그림이라고 성능 기준을 재정의한 후 다시 이미지를 평가해보았고 그 결과, 문장에서 불용어를 제거한 후 각 문장 성분을 쉼표로 구분하는 경우가 가장 좋은 평가를 받았습니다. 이를 통해 **유의미한 문장 성분은 최대한 살리되, 각 어절들을 따로 인식하도록** 하는 것이 모델의 이해를 돕는다는 것을 깨닫게 되었습니다.

- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?

  CLIP Score 나 FID Score 와 같은 **정량 평가**의 코드를 구현해 실험을 진행했기 때문에 일정 수준의 **객관적 성능을 입증**할 수 있었습니다. **기능별로 코드의 모듈화**를 진행하여 리팩토링을 할 때 구조를 조금만 변형하면 되는 식으로 손쉽게 버전을 업데이트할 수 있도록 했습니다. 문장 변형 실험을 통해 **유의미한 문장 성분은 최대한 살리되, 각 어절들을 따로 인식**하도록 하는 것이 모델의 이해를 돕는다는 것을 깨닫게 되었습니다. AI 서비스 개발에 있어서 **명확하고 실현가능한 문제 정의와, 그에 맞는 데이터셋 구축**을 순차적으로 수행하는 것이 이후 프로세스 전반과 모델링 성능에 지대한 영향을 준다는 것을 깨달았습니다.

- 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

  이전에는 하나의 큰 기능을 완성하는 것을 하나의 업무로 인식했었다면 이번에는 최대한 잘게 나누어 각각의 태스크를 차근차근 구현하였습니다. 함수화 및 모듈화한 기능들이 다른 곳에 쓰이기도 하기 때문에 **재사용성**이 커지는 장점이 있었습니다. 그리고 한 파일 내 코드 길이도 줄어들고 더불어 유의미한 네이밍을 사용함으로써 코드의 **가독성**이 훨씬 높아졌습니다.

개인회고: 허치영

이번 대회의 목표

별도로 제공된 베이스라인 코드 없이 처음부터 원하는 형태의 모델을 제작하고 fine-tuning 후 huggingface hub 에 올리는 것이 목표였습니다. 또한 충분히 실생활에 사용되어도 쓸 만한 수준의 성능을 나타내도록 학습하는 것이 목표였습니다. 모델 외적인 부분으로는 product serving 의 전체적인 흐름을 체험해보고, 흥미에 따라 앞으로의 진로 방향과 학습 방향을 정하고자 했습니다.

목표 달성을 위해 한 것

Dialogue summarizaion 모델의 동작 코드 제작을 위해 huggingface github 에 업로드 되어있는 예시 코드를 참고하여 모델 학습에 필요한 코드만을 담은 대화 요약이 가능한 모델 학습 코드를 작성했습니다.
실제로 충분히 사용할 만한 대화 요약 모델을 학습하고자 했고, 모델이 생성한 요약 문장과 정답 요약문 사이의 평균 cosine similarity 가 0.6 이상이 되도록 학습을 진행했습니다.
최종 프로젝트를 진행하며 제공된 강의를 듣고, 어떤 작업을 할 때 가장 재밌었는가에 대해서 생각해보면 모델 제작이었습니다. 현재 가장 재미를 느낀 부분은 논문을 읽고, 모델을 만들고 학습시켜보는 과정이었기에, 이후 연구직으로 취직하고 싶다는 마음을 갖게 되었습니다.

깨달은 것들

- **프로젝트 제작 시 모델링은 상상 이상으로 적은 부분을 차지한다.**

  데이터셋 제작과 검색, 프론트엔드, 백엔드의 비중이 모델링에 비해 훨씬 큰 것을 느낄 수 있었습니다. 이번 프로젝트를 진행하는 동안 모델링에 초점을 맞춰 작업을 했으나 실제로 작업할 양이 더 많았던 것은 프론트/백엔드 작업과 데이터셋 제작 단계였습니다. 모델링 이외의 부분에서는 지식이 깊지 않아 큰 역할을 하지 못해 아쉬웠습니다. 모델링에 대한 공부만을 할 것이 아니라 프로젝트 진행에 필요한 다른 부분들에 대한 학습도 병행해야 한다는 다짐을 하게 되었습니다.

- **주제에 맞는 데이터셋의 유무, 데이터셋의 양과 질에 따라 모델의 성능이 크게 차이 난다.**

  Text to image 모델을 학습할 때 대화 요약과 이미지의 쌍으로 이루어진 데이터셋이 없어서 pixabay 의 사진을 크롤링해서 학습시켰습니다. 생각보다 좋지 않은 성능을 나타내서 멘토님의 조언대로 대화 요약문, 이미지의 쌍으로 이루어진 데이터셋을 제작하고자 했고, 그 결과 눈에 띄게 좋아진 성능을 얻을 수 있었습니다.
  아쉬웠던 것은 충분히 긴 기간을 두고 좀 더 많은 데이터셋을 수집하거나 데이터의 품질을 향상시키지 못했다는 것입니다. 조금 더 긴 기간동안 작업을 진행했다면 더 좋은 성능을 얻어낼 수 있을 것이라 생각합니다. 이후 진행하게 될 프로젝트에서는 데이터셋 제작과 검색에 더 많은 시간을 할애해야 한다는 다짐을 할 수 있었습니다.

- **논문은 항상 정답이 되지 않으며, 내가 필요한 정보를 선택해 작업에 적용할 수 있어야 한다.**

  대화 요약 모델의 정량적 성능 평가 지표 설정에 있어서 어떤 지표를 선택해야 할지에 대한 깊은 고민을 했습니다.
  ROUGE 는 한국어 대화 요약 모델에 있어서 동의어와 한국어 접사로 인한 문제 때문에 제대로 성능 평가를 하지 못해 새로운 지표를 사용하기로 했고, RDASS 의 사용을 고려했습니다. 원문서, 정답 요약, 모델 생성 요약 셋 사이의 유사도를 이용하는 RDASS 는 대화 요약의 특성에 잘 맞지 않는다는 생각을 했고, RDASS 논문의 아이디어를 참고해 정답 요약과 모델 생성 요약 사이의 코사인 유사도만을 이용해 대화 요약 모델의 성능을 정량적으로 평가했습니다.
  코사인 유사도가 대화 요약 모델 성능 평가에 걸맞은 지표인가에 대한 연구 결과가 없었기에 높은 신뢰성을 갖지 못한 것이 아쉬웠습니다. 논문의 방법이 반드시 정답이 되지 않는다는 것을 알 수 있었고, 하려는 일에 맞게 논문의 아이디어를 변형시키거나, 필요한 아이디어만을 선택해서 가져오는 등 비판적 사고를 통해 정보를 선택 취합 하는 것이 필요하다는 것을 알게 되었습니다.

개인회고: 이보림

이번 프로젝트 목표

- 단순 기술 응용을 위한 프로젝트가 아닌 실생활의 문제를 해결하는 프로젝트 완성하는 것이다.

팀에서의 내 역할 및 기여

- 아이디어 회의 시 다양한 의견을 제시하였으며 채택되었다.
- 대화요약 텍스트를 이미지로 생성하는 부분을 맡았다.
- 그 중에서 데이터셋 구축, 모델 fine-tuning, infernece, Docker를 진행하였다.
- 프로젝트 마무리 단계에서 데이터 셋을 다시 구축하고 학습을 시켰는데, 이로 인해 많은 성능 향상이 있었다.

전과 비교해서 새롭게 시도한 변화

- Multi-modal task를 처음으로 시도해보았다.
- Pytorch Lightning을 이용하여 학습시켜보았다.
- 프로젝트에서 도커 이미지를 만들고 GCP의 Cloud build, run으로 실행시켜보았다.

한계 및 아쉬웠던 점

- Multi-modal 모델을 많이 실험해 볼 시간이 부족해서 아쉬웠다.
- Text2image에서 한국어를 이용한 pre-trained model이 없어서 번역을 해야한다는 점이 아쉬웠다.
- 대화 내용 내보내기 기능이 PC 카카오톡에만 있어서 PC로만 가능하다는 한계가 있다.
- 이미지를 생성한 후에 사용자가 직접 다운받아 변경하는 것이 아닌 사용자가 원하는 이미지 선택 후 자동 업데이트 해줬으면 더 의미가 있었을 것 같지만 불가능 하다는 한계가 있다.
- 도커가 컨테이너 생성 및 GCP의 Cloud Run에 배포하였지만, 학습시킨 GPU 환경에서 도커 사용 못하는 한계와 test 할 때 CPU환경에서는 매우 느려 결국 활용하지 못해서 아쉬웠다.

대회를 통해 알게 된 점

- 무엇보다 데이터의 영향력에 대해 느낄 수 있었다.
  처음에는 대화요약과 이미지생성 부분을 매우 독립적으로 생각하여 이미지 저작권과 텍스트-이미지 쌍에만 초점을 두어서 명확한 제목과 이미지가 있는 Pixabay 에서 데이터를 가져와 데이터셋을 구축하였다. 나중에서야 대화요약과 이미지생성은 하나의 프로젝트 안에 세부 task 이며 연결되어 있기에 이들의 도메인을 맞춰주는 중요성을 알게되었다. 프로젝트 마무리 할 시간에 급하게 대화요약팀의 정답 요약문을 이용하여 텍스트 번역, 전처리와 구글 검색 크롤링으로 데이터셋을 새로 모아 학습도 다시 진행하였다.
  이를 통해 대화요약과 관계없는 데이터인 Pixabay 를 사용할 때, 모델 입장에서는 새로운 문제를 해결하라는 것처럼 느꼈을 수도 있다고 생각되었다. 그리고 번역, 관련된 텍스트-이미지 쌍의 정확도 등 데이터가 생성되는 과정에서 노이즈가 들어갈 수도 있기에 텍스트-데이터쌍이 더 확실한 Pixabay 가 좋다고 생각하였는데, 이러한 중간에 노이즈가 들어가더라도 도메인을 맞춰주는 것이 얼마나 더 중요한 지 알게 되었다.
- 모델 입장에서 이 데이터는 어떻게 해석할까를 생각해 볼 수 있었다.
  Pixabay 데이터셋 구축 시, 카톡방 대표 이미지는 실제 사용 시 작게 표시되니까 일러스트나 패턴처럼 명확히 구분할 수 있는 것이면 좋겠다라고 생각했다. 그래서 일러스트와 패턴, 경치 위주로 데이터를 모았으며 당연하게도 검은 배경의 이미지가 적지 않은 비중을 차지하였다. 이후 학습된 모델을 테스트 할 때, 아예 검은 바탕의 이미지가 많이 생성되었다. 그래서 이 모델이 pre-trained 될 때 학습된 데이터와 내가 모은 데이터와 비교하였으며, 왜 검은색 이미지가 나올까 해석하는 시간을 가질 수 있었다.
- 프로젝트 방법론의 중요성을 알게되었다.
  내가 낸 아이디어기는 했지만, 막상 시작하려니 막막한 부분들이 많았다. 문제가 잘 풀리지 않아서 팀의 분위기와 의욕이 저하되면 어떡하자 고민하였는데, 세부 문제로 나누고 차근차근 해결하고 스크럼 회의마다 각자 어려운 점을 나누다 보니 끝까지 프로젝트를 잘 완성할 수 있었다. 만약 각자 완벽하게 역할을 나누고 진행하였다면 이렇게 까지 진행이 잘 되지 않았을 것 같다.
- 이번 프로젝트를 하면서 새로 설치해야하는 것들이 많아 환경 맞추는 것에 어려움이 있었다. 이로 인해 도커 사용의 필요성을 느꼈다.