



Dialogue Summarization

일상 대화 요약

Team 12 : simple 12조

목차

- 1 팀 소개
- 2 진행 과정
- 3 대회 소개
- 4 데이터 전처리
- 5 모델 학습
- 6 대회 결과
- 7 인사이트 및 회고
- 8 Q&A

1 팀원 소개



박범철

- 팀장, 발표
- Modeling



김나리

- Pre-processing
- Modeling



최윤설

- Pre-processing
- Modeling



조용중

- Pre-processing
- Modeling

2 진행 과정



1

대회 분석

2

BaseLine 분석, Pre-processing

3

모델 선정 (Bart → T5 → llama)

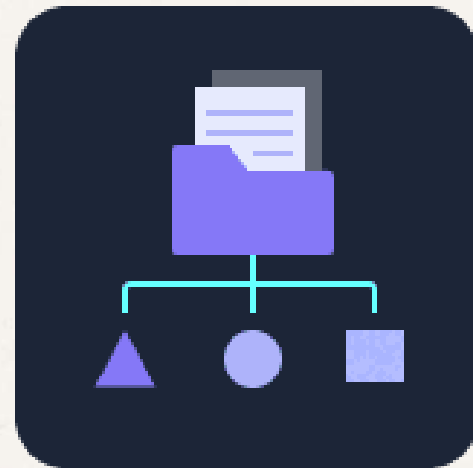
4

모델 학습 기간 설정

5

모델 학습

2 지난 대회 회고



구글 엑셀을
사용하여 실험
기록 공유

대회 날짜에
맞춘 시간 배분

같은 모델, 같은
튜닝 사용 자제

데이터 버전
맞추기

2 실험 기록 공유



Google
스프레드시트

데이터 & 모델
리더보드 제출

A	B	C	D	E	F	G	H	I	J	K	L	M
실험날짜	실험자	모델	비고	train loss	eval 점수				리더보드 점수			
					loss	rouge1	rouge2	rougeL	final_result	rouge1	rouge2	rougeL
2024년 08월 30일	yoonseol choi	digit82/kobart-summarization	baseline code + special tokens (best train loss)	0.3997	0.56373	0.37692	0.13836	0.36103	41.9809	0.5136	0.3175	0.4283
2024년 08월 30일	김나리	digit82/kobart-summarization	baseline code + special tokens		0.5609	0.38185	0.14523	0.36763	41.6822	0.5098	0.3161	0.4246
2024년 09월 01일	yoonseol choi	digit82/kobart-summarization	baseline code + special tokens + modify max len		0.58218	0.38521	0.1437	0.36977	41.6581	0.5113	0.316	0.4224
2024년 09월 02일	김나리	digit82/kobart-summarization	고감자 + baseline code + special tokens		0.577357	0.303733	0.079945	0.295802	41.1863	0.5078	0.3112	0.4166
2024년 09월 02일	yoonseol choi	digit82/kobart-summarization	baseline code + special tokens + 자/모음 전처리 (best eval loss)	0.4394	0.558012	0.384407	0.145014	0.369641	41.7765	0.5121	0.3161	0.4251
2024년 09월 02일	yoonseol choi	digit82/kobart-summarization	baseline code + special tokens + 자/모음 전처리 (best train loss)	0.3989	0.562177	0.388444	0.147888	0.374012	42.1839	0.5163	0.321	0.4282
2024년 09월 02일	박범철	digit82/kobart-summarization	baseline code + special tokens	0.4012	0.56812	0.38132	0.14123	0.36199	41.9809	0.5136	0.3175	0.4283
2024년 09월 03일	김나리	digit82/kobart-summarization	baseline+special+1024/512max_len+batch 4(best train loss)	0.1931	0.60348	0.30548	0.0812	0.29747	41.4494	0.5095	0.3138	0.4201
2024년 09월 03일	김나리	digit82/kobart-summarization	baseline+special+1024/512max_len+batch 4(best val loss)	0.2751	0.57653	0.31156	0.0813	0.30428	40.7299	0.5044	0.3065	0.411
2024년 09월 04일	yoonseol choi	digit82/kobart-summarization	D8 + min_length_64						29.4991	0.3745	0.2176	0.2928
2024년 09월 04일	yoonseol choi	digit82/kobart-summarization	D8 + min_length_32						37.9575	0.4723	0.2837	0.3827
2024년 09월 04일	yoonseol choi	digit82/kobart-summarization	D8 + length_penalty_1.0						41.4723	0.5089	0.312	0.4233
2024년 09월 04일	yoonseol choi	digit82/kobart-summarization	D8 + length_penalty_0.5						41.737	0.5104	0.3151	0.4265
2024년 09월 04일	yoonseol choi	digit82/kobart-summarization	D8 + length_penalty_0.25						41.928	0.5126	0.3166	0.4287
2024년 09월 04일	yoonseol choi	digit82/kobart-summarization	D8 + length_penalty_0.25 + num_beams 5						41.9264	0.5132	0.3166	0.428
2024년 09월 04일	김나리	lcw99/t5-large-korean-text-summary	각 화자별 요약하는 프롬프트 사용.	1.4215	1.3879	0.2028	0.0615	0.19632	38.4511	0.4743	0.2801	0.3991
2024년 09월 04일	yoonseol choi	digit82/kobart-summarization	D8 + no_repeat_ngram_size 2 -> repetition_penalty 1.2						40.9218	0.4998	0.3088	0.4191
2024년 09월 04일	김나리	digit82/kobart-summarization	chunk + leng_penalty_2.0	0.281	0.55962	0.38262	0.13978	0.36547	40.6414	0.499	0.3065	0.4138
2024년 09월 05일	Cho	lcw99/t5-large-korean-text-summary	output test + 1 epoch	1.6605	1.2468	0.2391	0.0821	0.2295	42.3105	0.5184	0.3174	0.4336
2024년 09월 05일	박범철	lcw99/t5-large-korean-text-summary	epoch 11	1.5586	1.1239	0.2684	0.1006	0.2589	43.6066	0.5322	0.3331	0.443
2024년 09월 05일	Cho	lcw99/t5-large-korean-text-summary	7 epoch	0.9171	1.1204	0.2681	0.098	0.2577	43.9477	0.535	0.336	0.447
2024년 09월 05일	Cho	lcw99/t5-large-korean-text-summary	7 epoch + inference parameter tuning + temperature = 0.3						37.8217			
2024년 09월 05일	Cho	lcw99/t5-large-korean-text-summary	7 epoch + inference parameter tuning : "generate_max_length": 512, #256, "num_beams": 5, #4						44.1122	0.5355	0.3389	0.449

3 대회 소개

Dialogue Summarization

여러 인물들이 나눈 대화 요약

평가 기준

- ROUGE-1-F1, ROUGE-2-F1, ROUGE-L-F1 세 가지 metric을 사용해 최종 점수 산출
- Multi-Reference Dataset의 특성에 맞춘 평가 방법: 여러 정답 요약 문장 중 3개를 비교하여 평균 점수를 계산함
- 랜덤하게 선택된 요약 문장의 평균 점수가 약 70점임

이번 대회 전략

- 각종 모델의 large모델을 활용
- Large 모델의 inference 파라미터 수정

4 데이터 전처리

- 오타자 수정 (철자 오류 등 수정)
- 마스킹 처리 (Special token 적용)
- 자/모음으로만 구성된 문자열 제거 (정규식 활용)

Dialogue 오타자

```
replacements = {
    'ㅋㅋ': '웃기다', 'ㅇ로': '으로',
    '제ㅏ': '제가', '표알': '알',
    'ㄷ거': '거',
    '##': '#', '회사 #에서': '회사에서',
    '#작은': '#Person2#: 작은', '#여기서': '#Person1#: 여기서',
    '#나': '#Person2#: 나',
    '#페리에와': '#Person1#: 페리에와',
    '#샐러드용': '#Person1#: 샐러드용',
    '#어디': '#Person1#: 어디',
    '#잠깐만요': '#Person1#: 잠깐만요',
    '#하지만': '#Person1#: 하지만',
    '#사람1만기': '#Person1#: 만기',
    '#PhoneNumber이고': '#PhoneNumber#이고', '#Person1:': '#Person1#:',
    '#Person2:': '#Person2#:', '#Person#': '#Person2#:', '사람1#': '#Person1#:',
    '#고객님:': '#Person2#: 고객님',
    '선생님: ': '', '로저스 씨: ': '',
    '남자: 아악.': '', '남자: 고마워.': ''
}

df['dialogue'] = df['dialogue'].replace(replacements, regex=True)
```

Summary 오타자

```
if 'summary' in df.columns:
    summary_replacements = {
        '사람1#': '#Person1#', '사람2#': '#Person2#', '#사람1#': '#Person1#'
    }
    df['summary'] = df['summary'].replace(summary_replacements, regex=True)
```

Special token 적용

```
1 from transformers import AutoTokenizer
2
3 # 토큰나이저 로드
4 tokenizer = AutoTokenizer.from_pretrained("gogamza/kobart-base-v2")
5
6 # 특별 토큰 추가
7 special_tokens = list(set.union(*masked_info))
8 special_tokens_dict = {'additional_special_tokens': special_tokens}
9 tokenizer.add_special_tokens(special_tokens_dict)
10
11 print("추가된 특별 토큰:", tokenizer.additional_special_tokens)
12 print("추가된 특별 토큰 ID:", tokenizer.additional_special_tokens_ids)
```

Python

config.json: 0%| | 0.00/1.36k [00:00<?, ?B/s]

You passed along `num_labels=3` with an incompatible id to label map: {'0': 'NEGATIVE', '1': 'POS

tokenizer.json: 0%| | 0.00/682k [00:00<?, ?B/s]

added_tokens.json: 0%| | 0.00/4.00 [00:00<?, ?B/s]

special_tokens_map.json: 0%| | 0.00/112 [00:00<?, ?B/s]

You passed along `num_labels=3` with an incompatible id to label map: {'0': 'NEGATIVE', '1': 'POS

추가된 특별 토큰: ['#Email#', '#Person5#', '#DateOfBirth#', '#SSN#', '#CarNumber#', '#Person#', '#
추가된 특별 토큰 ID: [30000, 30001, 30002, 30003, 30004, 30005, 30006, 30007, 30008, 30009, 30010,

5 모델 학습 – koBart

시도한 방법들

1. 번역후 요약 시도

과정: 대화를 영어로 번역 후, 영어 BART 모델로 요약하고 다시 한국어로 번역하는 방식

문제점: 번역에서 발생하는 오류와 요약 과정에서 발생하는 오류가 중첩되어 성능 저하

2. 강화학습 알고리즘을 적용한 모델 업데이트

방법: ROUGE 점수를 보상 신호로 활용하여 모델을 강화학습 알고리즘으로 학습

목표: 모델이 더 높은 ROUGE 점수를 내는 방향으로 학습되도록 설계

3. K-Fold

방법: 데이터를 여러 Fold로 나눠 교차 검증 시행

4. K-Fold + 강화학습

방법: K-Fold + 강화학습을 결합하여 더 견고한 모델로 학습 시도

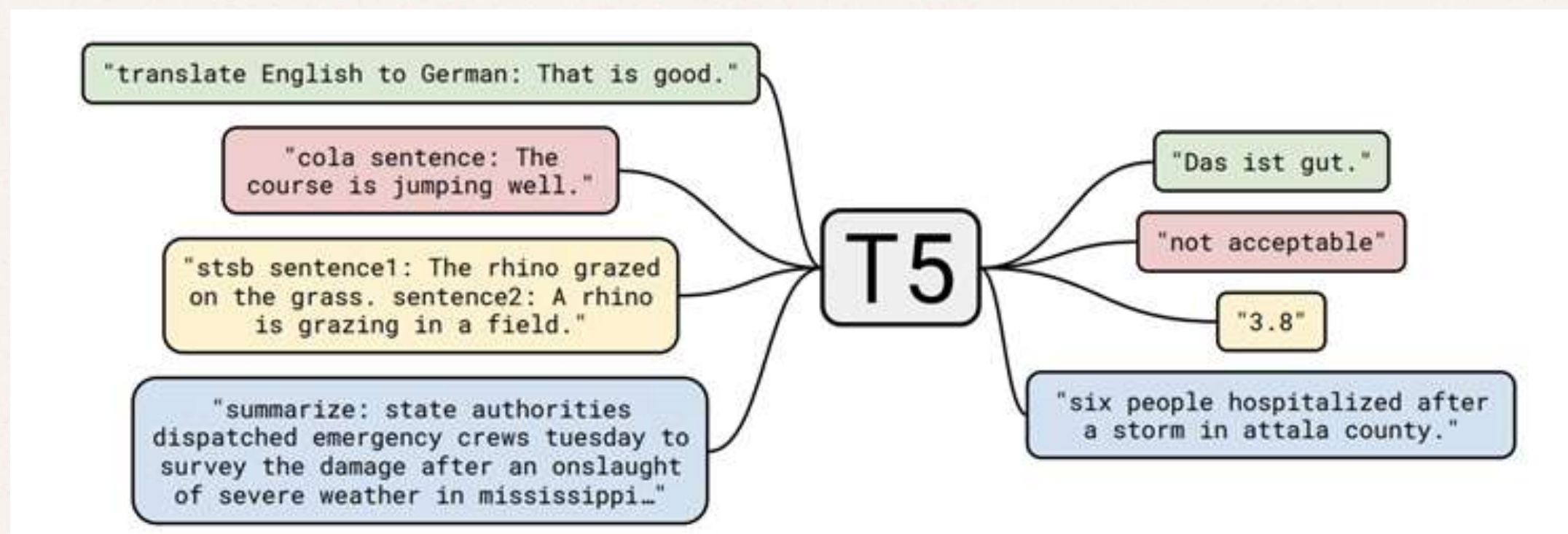
1~4 과정을 시행한 결과
Baseline을 넘지 못했다.

오히려 Baseline에서
max_length를
1024/512로 수정한 방법이
더 점수가 좋았다.

5 모델 학습 – T5_large

T5 (Text-to-Text Transfer Transformer)

- 모든 NLP 작업을 텍스트 입력 -> 텍스트 출력 형태로 통일
- Seq2Seq 아키텍처 : 인코더가 입력 텍스트를 벡터로 변환 후, 디코더가 이 벡터를 바탕으로 출력 텍스트 생성
- 성능 : 다양한 NLP 벤치마크에서 최고 수준의 성과를 보여줌 적은 데이터로도 높은 성능 발휘 (Few-shot/Zero-shot 학습)



5 모델 학습 - T5_large

lcw99/T5 모델로 시도한 방법

1. CUDA 메모리 오류 발생

문제 : 모델이 크기에 서버에서 감당 못하여 메모리 부족 오류 발생

해결 : 오류 발생 시 check포인트를 만들어 오류 난 곳부터 다시 학습 진행

메모리 오류 Exception 처리

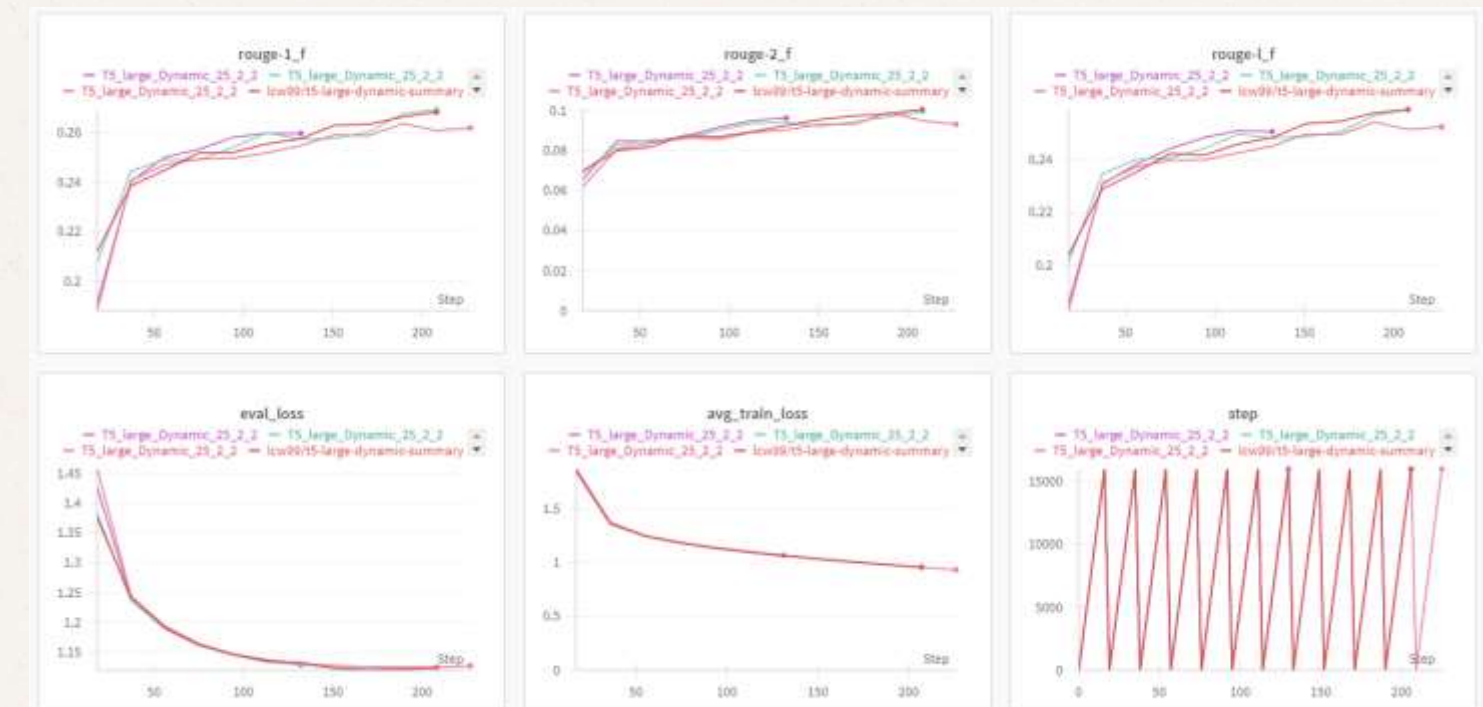
```
# CUDA Out of Memory Exception 처리 (계속 에러가 나는 경우 kill -9 로 백그라운 파이썬 강제 처리 해야함)
while retry_count < max_retries:
    try:
        train_and_save(config)
        break
    except RuntimeError as e:
        if "out of memory" in str(e):
            retry_count += 1
            logger.warning(f"CUDA out of memory error occurred. Attempt {retry_count} of {max_retries}.")
            torch.cuda.empty_cache()
            if retry_count == max_retries:
                logger.error("Max retries reached. Exiting.")
                raise
        else:
            logger.error("Unexpected error occurred.", exc_info=True)
            raise
```

2. 모델 성능 최적화

파라미터 수정 및 inference : inference 단계에서

파라미터를 수정하여 최종적으로 최고 점수 기록

t5 large mod...ng ll		0.5334	0.3403	0.4513	44.1650
		0.5172	0.3081	0.4214	41.5561



5 모델 학습 - llama3

Beomi/Llama-3-Open-Ko-8B

- 사전 학습 데이터: 중복 제거된 60GB 이상의 공개 텍스트 데이터로 학습 됨
- 토큰나이저: 새로운 Llama3 토큰나이저를 사용해 177억 개 이상의 토큰으로 사전 학습 진행. 이전 Llama2-Ko 토큰나이저보다 더 많은 토큰 사용
- 특징: 대용량 데이터를 기반으로 다양한 언어 작업에서 뛰어난 성능을 발휘

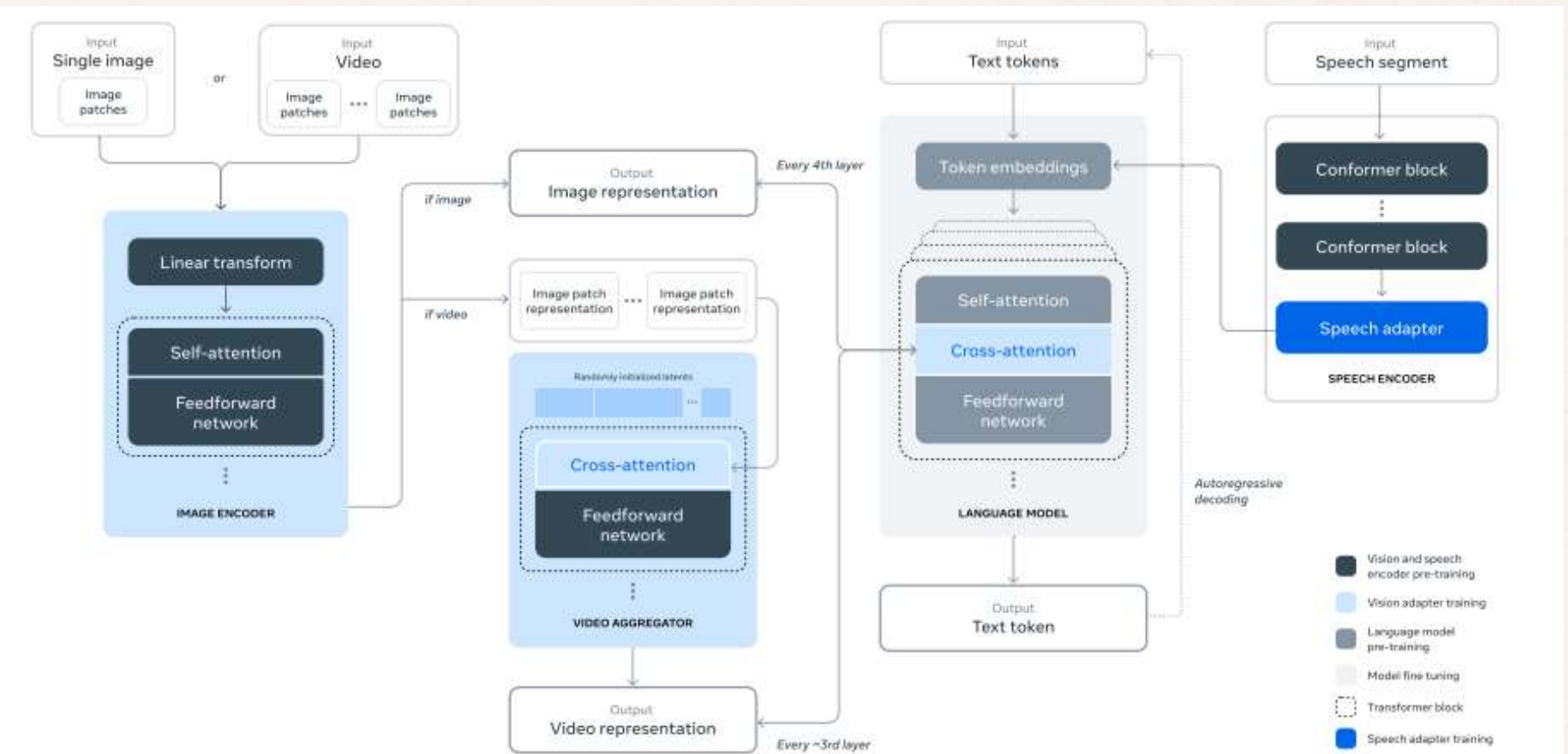


Figure 28 Illustration of the compositional approach to adding multimodal capabilities to Llama 3 that we study in this paper. This approach leads to a multimodal model that is trained in five stages: **(1)** language model pre-training, **(2)** multi-modal encoder pre-training, **(3)** vision adapter training, **(4)** model finetuning, and **(5)** speech adapter training.

5 모델 학습 – llama3

Llama3 모델로 시도한 방법

1. CUDA 연산 능력 확인

GPU의 CUDA 연산 능력이 8 이상일 경우, 고성능 GPU에서 Attention 메커니즘을 선택하고, torch 데이터 타입을 bfloat16으로 설정하여 메모리 사용량을 줄이면서 계산 정확성 유지

2. LoRA (Low-Rank Adaptation)

PEET(파라미터 효율적 미세 조정) 기법인 LoRA를 사용해 모델의 일부 파라미터만 조정, 컴퓨팅 자원과 메모리 사용량 크게 줄임

3. QLoRA (Quantized LoRA)







LoRA에 4비트 양자화(Quantization) 기법을 추가해 더 적은 메모리와 자원을 사용하면서도 비슷한 성능을 유지

4. SFTTrainer

지도학습 방식으로 Llama3 모델을 효율적으로 미세 조정. 대규모 언어 모델의 학습 과정을 쉽게 관리

6 대회 결과

T5-large 모델을 활용하여 inference 튜닝을 통해 최고 점수 달성
대회 최종 순위 5위로 마무리

순위	팀이름	팀멤버	rouge1	rouge2	rougeL	final_result	제출횟수	최종 제출
내등수 5	12조		0.5172	0.3081	0.4214	41.5561	60	18h
1	NLP 11조 🏆	 최지	0.5426	0.3442	0.4525	44.6438	77	2d
2	6조 🏆	 전승	0.5367	0.3382	0.4504	44.1763	36	1d
3	2조 🏆	 선호 di	0.5196	0.3111	0.4270	41.9213	83	20h
4	NLP7 🏆		0.5140	0.3154	0.4223	41.7251	81	16h
5	12조 🏆		0.5172	0.3081	0.4214	41.5561	60	18h

7 인사이트 및 회고 – 아쉬운 점

박범철

시간이 많을 때 T5 모델 inference 튜닝한 모델을 돌려 봐야 했는데 계속 모델을 못 돌렸던 것이 아쉬웠다.

김나리

개인적으로는 계속 KoBART만을 Fine-tuning 하려고 노력했는데, 결과적으로는 잘 되지 않았다. 데이터 증강을 시도하다가 하지 않았는데 그게 너무 아쉬웠다.

최윤설

- Baseline에서 사용한 모델 외 타 KoBART 모델도 사용해보고 num_beams를 조정해보거나 length_penalty 및 repetition_penalty 값을 추가해도 KoBART 모델로는 최고의 성능을 낼 수 없어서 좀 아쉬웠다.
- Llama3 모델 사용 시 CUDA Memory 오류가 자주 발생하여 답답했다.

조용중

7 인사이트 및 회고 – 시도해보고싶은 점

박범철

T5_large 모델에
집중하느라 Llama3
모델을 같이 못하여 좀더
Llama3에 대해 알고
시도해보고싶다.

김나리

다른 조원들이 Llama나
T5등 fine_tuning 하면서
많은 것을 배우신 것
같았다. 대회는 끝나지만,
남은 온라인 수업과 함께
요즘 유행하는 모델들을
공부하고 직접 다루고 싶다.
그렇지만 KoBART만큼은
정말 많이 알고가서
뿌듯하다.

최윤설

학습 데이터 셋에 주어진
'topic'을 사용하지
않았는데 BERT모델을
활용해 topic 분류 후 각
topic마다 vocabulary를
활용하여 대화를
요약해보는 것

조용중

7 인사이트 및 회고 – 궁금한 점

Llama3

문제 : 1 epoch 출력에서 Special token 누락되었음.

확인한 사항

- Tokenizer의 vocab 확인
- 모델의 resize_token_embeddings 설정 점검
- Special token 관련 default parameter 설정을 모두 확인

궁금한 점

- 이 모든 부분을 점검했음에도 special token이 왜 빠져 있는지 아직 명확한 이유를 찾지 못해 의문이 남음.



감사합니다.

Thank You
