NLP 경진대회 3조

김태한, 권혁찬, 김소현, 문정의, 이현진

Content

01

팀원 소개

팀원 소개

이름	김태한	권혁찬	김소현	문정의	이현진
역할	**조장**	조원	조원	조원	조원
전공	응용통계학 인공지능 (석사)	선박기관시스템 공학과	컴퓨터공학과 (석사)	전자계산학과	산업경영공학과
관심사	Computer Vision, NLP, 추천 시스템 OTT, 문화산업 (책, 웹툰, 영화 등등), 쇼핑 등	기계 시스템 고장진단, 시계열 예측, LLM, MLops	Computer Vision, NLP, 최적화, 경량화	ML, 강화학습	Computer Vision, NL, 금융 데이터 및 물류 산업 데이터

02

대회 소개

대회 소개

대회 개요

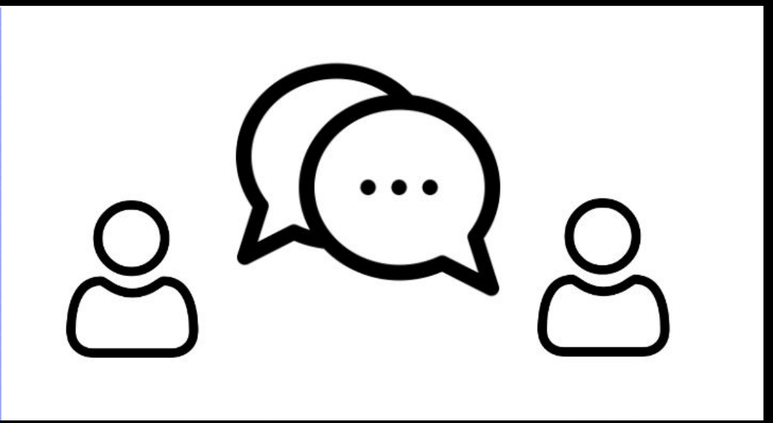
Dialogue Summarization | 일상 대화 요약

학교 생활, 직장, 치료, 쇼핑, 여가, 여행 등 광범위한 일상 생활 중 하는 대화들에 대해 요약합니다.

#비공개대회 #UpstageAILab1기 #NLPAdvanced

D - 1 | 2024.03.08 ~ 2024.03.20 19:00

16팀



목표 : 일상 생활 속 대화 Data를 1개의 문장으로 요약하기

제공된 회의, 일상 대화 등 다양한 주제에서 나누는 '대화문', 그리고 그 대화를 요약한 '요약문'을 이용하여 대화를 요약하는 모델의 성능을 높인다.

평가 지표

Rouge

모델의 요약본을 참조 요약본과 비교해서 점수 계산
대표적인 요약, 번역 task 평가 지표

- Rouge-N : n-gram의 중복 여부를 기준으로 Recall, Precision 계산
- Rouge-L : LCS를 이용해서 가장 길게 매칭되는 문자열 측정

Final Score

Rouge-1-F1 + Rouge-2-F1 + Rouge-L-F1의 평균합

ROUGE-N

$$\text{Recall} = \frac{\text{Gold와 Pred의 겹치는 N-gram의 수}}{\text{Gold의 N-gram의 수}} \qquad \text{Precision} = \frac{\text{Pred와 Gold의 겹치는 N-gram의 수}}{\text{Pred의 N-gram의 수}}$$

ROUGE-L

$$\text{Recall} = \frac{\text{가장 긴 공통부분 문자열의 길이}}{\text{Gold의 1-gram의 수}} = \frac{N_{(\text{under the bed})}}{N_{(\text{the, cat, was, under, the, bed})}} = \frac{3}{6}$$

$$\text{Precision} = \frac{\text{가장 긴 공통부분 문자열의 길이}}{\text{Pred의 1-gram의 수}} = \frac{N_{(\text{under the bed})}}{N_{(\text{under, the, bed, there, was, the, cat})}} = \frac{3}{7}$$

$$\text{Final Score} = \frac{\sum_i^N \boldsymbol{ROUGE-1-F1}\,(pred, gold_i)}{N} + \frac{\sum_i^N \boldsymbol{ROUGE-2-F1}\,(pred, gold_i)}{N} + \frac{\sum_i^N \boldsymbol{ROUGE-L-F1}\,(pred, gold_i)}{N}$$

대회 소개

데이터 소개

Train (12,457개)

fname : input data의 이름(train_n)
dialogue : input data, 2명의 사람이 약 5 ~ 6문장씩 발화한 내용
특수 토큰을 위해 #Person1#, #Person2#, /n으로 구분되어 있음
summary : dialogue 발화 내용을 요약한 1~2문장으로 구성된 참조 요약본
topic : dialogue 발화내용의 주제

Test (499개)

fname : input data의 이름(test_n)
dialogue : input data, 2명의 사람이 약 5 ~ 6문장씩 발화한 내용
특수 토큰을 위해 #Person1#, #Person2#, /n으로 구분되어 있음

03

Experiments

Experiments

Data Augmentation strategy

항목	내용
EasyDataAugmentation	• SR: Synonym Replacement, 특정 단어를 유의어로 교체 (작으면 성능 향상에 큰 도움) • RI: Random Insertion, 임의의 단어를 삽입 (클수록 좋음) • RS: Random Swap, 문장 내 임의의 두 단어의 위치를 바꿈 (≤ 0.2) • RD: Random Deletion: 임의의 단어를 삭제 (0.1)
EasyDataAugmentation	• SR: Synonym Replacement, 특정 단어를 유의어로 교체 • RI: Random Insertion, 임의의 단어를 삽입 • RS: Random Swap, 문장 내 임의의 두 단어의 위치를 바꿈 • RD: Random Deletion: 임의의 단어를 삭제 Dialogue의 바뀐 유의어와 summary의 원래 단어가 다르면 학습에 방해되리라 예상해서 SR은 제외
back translation	Papago를 이용해 back translation으로 데이터 증강 (+50%)
불용어 제거	• 감탄사 제거 • 의성어, 의태어 제거 • 그 외 접속사, 부사 등 개인적으로 만든 불용어 리스트 사용
특수문자 처리	• ‘!’, ‘!!’이나 ‘...’, ‘....’, ‘.....’ 같이 통일되지 않은 특수문자들 양식 통일 • 괄호 ( ), [ ] 안 내용 제거

Experiments

Pre-trained 모델 선정

Model	Result	비고
digit82 / kobart-summarization	• gogamza/kobart-base-v1를 pre-trained 모델로 활용 • Dacon 한국어 문서 생성요약 AI 경진대회의 학습 데이터로 fine-tuning • 기본 baseline 모델	
gogamza / kobart-summarization	• 한국 뉴스 요약 모델	• digit82/kobart-summarization대비 rouge2값이 낮음
jx7789 / kobart_summary_v2	• AIHub의 한국어 대화 요약, 한국어 SNS 데이터로 KoBART를 fine-tuning	
EbanLee / kobart-summary-v1	• KoBART 모델을 도서자료 요약 데이터로 fine-tuning	

- Huggingface에 업로드되어 있는 KoBART 위주로 모델 선정
- 그 외 T5나 KoT5를 적용하려는 시도도 해 보았으나 적용 오류로 해 보지 못함

(psyche/KoT5-summarization, google-t5/t5-small, KETI-AIR-Downstream/long-ke-t5-base-summarization, traintogpb/pko-t5-large-kor-for-colloquial-summarization-finetuned 등, batch size 오류 같은 문제 발생)

- 최종적으로 기본 baseline 모델로 실험 진행

Experiments

Hyper Parameter Tuning

Model	Hyper Parameter Tuning	Final result	Rouge1	Rouge2	RougeL
digit82 / kobart-summarization	baseline	41.4308	0.5079	0.3134	0.4216
	epoch : 20 → 50	41.0321	0.5038	0.3088	0.4184
	encoder_max_len : 512 → 1024 decoder_max_len : 100 → 1024	**41.9246**	**0.5128**	**0.3191**	0.4258
	special_token : 6 → 16	41.7098	0.5121	0.3166	0.4226
	weight_decay : 0.1 → 0.2	41.3788	0.5115	0.3109	0.4190
	lr_scheduler_type : cosine → linear	41.7599	0.5102	0.3166	**0.4260**
	gradient_accumulation_steps: 1 → 3	40.8555	0.5031	0.3077	0.4148
	num_beams : 4 → 5	41.7599	0.5102	0.3166	0.4260
	encoder_max_len : 512 → 1024 decoder_max_len : 100 → 512	41.4157	0.5099	0.3131	0.4195

Experiments

Data Pre-processing + Hyper Parameter Tuning

Model	Data Pre-processing / Hyper Parameter tuning	Final result	Rouge1	Rouge2	RougeL
digit82 / kobart-summarization	• baseline • 사용자 지정 불용어 제거	41.6326	0.5086	0.3172	0.4232
	• baseline • 감탄사 불용어 제거	41.3676	0.5086	0.3117	0.4206
	• encoder_max_len : 512 → 1024 decoder_max_len : 100 → 256 • 사용자 지정 + 감탄사 불용어 제거	41.6441	0.5103	0.3157	0.4234
	• decoder_max_len : 100 → 64 • 사용자 지정 + 감탄사 불용어 제거	41.7169	0.5121	0.3175	0.4219

- 문장 요소 중 없어도 구성과 의미에 영향이 없는 불용어를 제거하려는 시도
- 사용자 지정 불용어에는 의성어, 의태어, 부사, 접속사 포함
- 학습 데이터셋 기준 Encoder의 최대 input 개수는 2546개, Decoder의 최대 input 개수는 481개
  양단의 max_len을 조절했을 때, decoder는 너무 크지 않은 쪽이 더 좋은 성능을 보임
- Encoder와 Decoder의 max_len이 너무 크면 overfitting되는 경향이 존재

Experiments

Data Augmentation

Model	Data Augmentation	Final result	Rouge1	Rouge2	RougeL
digit82 / kobart-summarization	• baseline • EDA로 데이터 증강 12400 → 37265	40.5478	0.5013	0.3048	0.4103
	• baseline • EDA로 유의어 교체를 제외하여 online augmentation 방식으로 증강	41.3676	0.5086	0.3117	0.4206
	• baseline • Papago back translation (ko→en→ko) • Random sampling된 50%의 데이터 증강	41.7972	0.5114	0.3160	0.4264

- EDA(유의어 교체, 단어 삽입, 단어 교체, 단어 삭제)로 데이터 증강을 시도해 보았으나, 문맥을 파악해야 하는 복합어인 한국어의 특성상 유의미한 성능 향상을 보이지는 않았음
- back translation을 적용했을 때 기존 baseline보다 향상된 성능을 보임

강사님께 받은 피드백 및 의견을 정리해주세요.

Q. Encoder-only와 Decoder-only를 활용해서 summarization하는 방법은 직접 구현해야 할까요? 기존 모델을 사용하기엔 요약에 맞게 사전학습된 것이 아니기 때문에 성능을 내기엔 어려울까요?
A. **instruction tuning** 방식으로 별도의 코드를 작성해서 summarization task에 맞게 모델을 바꾸어야 한다. KoGEMMA 등을 사용해 볼 수 있다. 실험해 보아야 하겠지만, 큰 성능향상이 없을 수 있다.

Q. 한국어인 데이터를 영어로 번역해서 영어 summarization model을 사용하는 건 성능이 어떨까요? target과의 rouge 계산을 위해 다시 한국어 번역 과정을 거친다 해도 모델이 다르니 각자 사용했던 tokenizer의 vocabulary에 없는 단어 때문에 성능을 내기 어려울까요?
A. 질문에서 언급한 문제 때문에 rouge 성능은 나오지 않는다. 다른 단어로부터의 번역 모델의 성능에 의존하기도 하고, **tokenizer의 vocabulary에 없는 단어들이 많이 나와(unknown token)** rouge score를 계산하는 건 "같은 단어"를 맞춰야 하기 때문에 점수가 떨어질 수 있다.

강사님께 받은 피드백 및 의견을 정리해주세요.

Q. koEDA에서 SR, RD, RI, RS을 썼을 때의 성능이 크게 오르지 않는 경우 어떻게 해야 할까요?
A. 노이즈 추가를 했을 때 성능이 오를지 그렇지 않을지는 직접 실험을 해봐야 한다. 영어에는 꾸밈말이 많아서 전체 문장의 의미 생성에 큰 차이가 없다. 형태소 분석기로 쪼개도 하나 하나가 의미가 많다.
따라서 단어 단위로 바꾸면 의미 변화 차이가 크므로, koEDA나 AEDA보다는 back translation이 오를 가능성이 높다.
복합적 언어 한국어, 일본어, 몽골어 등의 언어는 문맥을 이야기하는게 중요하다. 영어는 단어 한두 개를 빼도 단어의 문맥이 바뀌지 않는 경우가 많지만 한국어는 조사 하나가 바뀌어도 의미가 크게 바뀐다.
단어 단위 증강보단 문장 단위 증강이 필요하며, 언어 모델 api를 이용하여 적용하는 경우가 많다.

Q. dialogue랑 summary 둘다 증강을 적용해야 하나요?
A. dialogue를 먼저 적용하고 summary를 나중에 적용해서 비교해본다. 모델 학습이 오래 걸리면 중간에 비는 시간을 잘 활용해야 한다.

04

결과

최종 순위 및 평가지표 결과

순위	팀 이름	팀 멤버	final_result	rouge1	rouge2	rougeL
7 (1 ▲)	NLP 3조		39.1958	0.4967	0.2858	0.3933

Final result 39.1958, Rouge1 0.4967, Rouge2 0.2858, RougeL 0.3933으로 최종 순위 7위 기록

05

경진 대회 진행 소감

경진 대회 진행 소감

문정의

다양한 모델과 하이퍼파라미터를 적용해 보았고, 하이퍼파라미터 변경시에 어떤 결과가 나타나는지 확인해 볼 수 있었다. 대부분의 경우 큰 성능향상이 나타나지 않아서 어려움이 있었고, T5 모델 적용하는 부분에 대한 이해도 및 참조 코드 부족으로 인해 적용해 보지 못한 것이 아쉬웠다. 멘토님의 학습데이터 증강이 필요하다는 조언을 좀 더 빨리 받아서 적용하지 못한 부분도 아쉽다.

김태한

베이스라인 코드의 성능이 높아서 성능의 향상이 어려웠다. 데이터 증강도 컴퓨터 비전과는 다르게 적용에 대한 이해와 적용 후 성능의 변화 추이에 대해서도 깊게 이해하지 못해서 아쉬웠다. BART 외의 다른 모델을 적용하려 시도했으나 성공하지 못해서 아쉬웠다. 추후에 huggingface의 코드 등을 확인하여 더 깊이 있는 이해를 시도해보고 싶다.
영어와 한국어의 서로 다른 특성을 이해하고 그에 맞게 데이터 증강과 모델을 적용하는 경험을 해야 겠다고 생각했다.

경진 대회 진행 소감

이현진

개인 사정으로 인해 대회 기간 많은 시간을 투자하지 못한 것이 아쉽다. 데이터를 살펴보고 모델에 적용할 여유가 없던 것이 가장 아쉬우며, 다양한 모델에 관해 생각할 여유가 없던 것도 아쉽다. 하지만 요약 모델을 강의에서 들은 내용과 함께 코드를 확인해볼 수 있어서 도움이 많이 된 것 같다.

김소현

처음 공부해 보는 NLP였고, 강의 내용을 이해하는 데에도 많은 시간을 투자했다. 어려웠지만 형태소 분석 등 언어학에 대해서도 익히고, 그런 언어학적인 분석이 프로그래밍으로 이어져 기계가 이해하도록 설계하는 방법을 배울 수 있는 유의미한 시간이었다. 대회에서 많은 시도를 해 보진 못했으나 자연언어 처리라는 분야에 흥미를 붙일 수 있게 되어 좋았다. 앞으로는 강의와 대회를 경험하며 얻은 지식들을 계속 공부하며 직접 코드도 설계해 보는 시간을 가질 계획이다.

References

KoEDA
https://github.com/toriving/KoEDA

Papago API, papago, selenium
https://developers.naver.com/docs/papago/README.md
https://papago.naver.com/
https://github.com/SeleniumHQ/selenium

Huggingface Korean Summarization Models
https://huggingface.co/models?pipeline_tag=summarization&language=ko&sort=trending

Q&A

감사합니다.