[기업과제 4] - 개인 보고서

9팀 강민지

1. 담당한 역할

1) [전처리] AI Hub에서 추가 추출요약 데이터 확보 후 모델 input 형태로 전처리
2) [전처리] CLOVA Summary API를 사용하여 sports_news_data에 대한 TRUE SUMMARY를 생성하고 추출요약 학습 데이터셋 구축, 모델 input 형태로 전처리
3) [Inference] 모델 실행하여 PREDICT SUMMARY 도출

2. "TRUE SUMMARY"를 생성한 방식 및 근거

본 프로젝트에서는 '추출요약 모델'을 생성하고자 하였는데, 모델 학습 및 평가를 위해서는 sports_news_data에 대한 정답 셋을 구축하는 작업이 필요하였다. 가장 좋은 방법은 여러 명의 사람이 객관적인 기준을 마련하여 직접 정답 셋을 구축하는 것이나, 짧은 프로젝트 수행 시간을 고려하여 네이버에서 제공하는 CLOVA Summary API를 사용하여 추출요약 정답 셋을 구축하기로 결정하였다. API를 사용하는 과정이 이미 학습되어 있는 특정 모델을 사용하여 결과를 얻어낸다는 점에서 TRUE SUMMARY를 생성하는 가장 이상적인 방법이라고 할 수는 없으나, CLOVA API는 네이버 뉴스 기사 요약봇 등에도 공식적으로 활용되고 있기 때문에, 프로젝트에 소요되는 시간 및 자원을 고려하였을 때 가장 전문적이고 객관적인 요약 결과를 얻을 수 있을 것이라 판단하였다.

CLOVA Summary API에서는 몇 가지 옵션(파라미터)을 지정할 수 있는데, 중요한 옵션으로는 model의 종류('general', 'news' 등)와 SummaryCount(Summary로 추출해낼 문장의 개수)가 있다. 5개 샘플 뉴스기사로 테스트해 본 결과, 'news'보다는 'general' 모델이 더 좋은 요약 성능을 보이는 것으로 판단되어 'general' 모델을 선택하고, SummaryCount는 3으로 지정하여 최종 요약문이 3개 문장으로 구성되도록 하였다.

3. 평가 지표 함수를 선택 또는 정의한 근거

본 프로젝트에서는 ROUGE-1 score를 사용하여 요약 텍스트 스코어를 계산하였다. ROUGE는 문서 요약, 기계 번역 등 모델의 성능을 평가하기 위해 일반적으로 사용되는 지표 중 하나이며, 모델이 생성한 요약문(PREDICT SUMMARY)과 정답(TRUE SUMMARY)에 등장하는 토큰에 기반하여, 몇 개의 토큰이 일치하는지를 기반으로 도출하는 평가 지표이다. ROUGE-recall과 ROUGE-precision, ROUGE-F1은 다음과 같은 수식으로 계산한다.

$$ROUGE_{recall} = \frac{\text{겹치는 토큰 수}}{\text{TRUE SUMMARY의 토큰 수}}$$

$$ROUGE_{precision} = \frac{\text{겹치는 토큰 수}}{\text{PREDICT SUMMARY의 토큰 수}}$$

$$ROUGE_{F1} = 2 \times \frac{\text{ROUGE}_{\text{recall}} \times \text{ROUGE}_{\text{precision}}}{ROUGE_{recall} + ROUGE_{precision}}$$

생성 요약 모델의 경우, PREDICT SUMMARY에서 의미적으로 원문과 비슷하지만 언어학적 형태가 다른 토큰이 많이 등장할 수 있으므로 토큰의 1:1 비교를 통한 점수를 계산하는 ROUGE score는 이상적이라고 할 수 없다. 그러나, 본 프로젝트에서는 추출요약 모델을 구성하였고 추출요약 task의 목표는 원문을 구성하는 n개의 문장 중에서 가장 중요한 정보를 담고 있는 k개의 문장을 추출해내는 것이다. 따라서 PREDICT SUMMARY에 사용된 토큰들이 TRUE SUMMARY에 사용된 토큰들과 "완전하게 일치하는지"를 평가하는 것이 필요하므로 ROUGE score를 사용하였을 때 비교적 객관적으로 추출요약 모델의 성능을 계산할 수 있을 것이라 판단하였다.

4. 모델 선정 배경 및 이유

Rank	Model	ROUGE-1	ROUGE-2	ROUGE-L	Extra Training Data	Paper	Code	Result	Year	Tags
1	HAHSum	44.68	21.30	40.75	×	Neural Extractive Summarization with Hierarchical Attentive Heterogeneous Graph Network			2020	
2	MatchSum	44.41	20.86	40.55	×	Extractive Summarization as Text Matching			2020	
3	NeRoBERTa	43.86	20.64	40.20	×	Considering Nested Tree Structure in Sentence Extractive Summarization with Pre-trained Transformer			2021	
4	BertSumExt	43.85	20.34	39.90	✓	Text Summarization with Pretrained Encoders			2019	
5	BERT-ext + RL	42.76	19.87	39.11	×	Summary Level Training of Sentence Rewriting for Abstractive Summarization			2019	

(출처: https://paperswithcode.com/sota/extractive-document-summarization-on-cnn)

영어 CNN/DailyMail Dataset을 사용하여 추출요약 task를 진행한 연구들 중에서 높은 성능을 기록하고 있는 모델로 MatchSum, BertSum이 있어 해당 모델을 한국어에도 적용시켜 보기로 결정하였다.

5. 모델 훈련 과정

- MatchSum[1]

[모델 정보]

- Pretrained Model: klue/bert-base
- Loss Function: BCELoss
- Optimizer: Adam
- Learning Rate: 1e-5
- Epochs: 5
- Evaluation Metric: Rouge-1

[훈련 과정]

[1] https://victordibia.com/blog/extractive-summarization-pytorch/에서 참고

1) sports_news_data.csv에 있는 기사 원문(CONTENT)에 대해 CLOVA Summary API를 사용하여 3문장으로 구성된 추출 요약문 정답셋을 구축한다.
2) AI Hub에서 추가적인 추출 요약문 데이터셋(https://aihub.or.kr/aidata/8054)을 확보하고, 이를 1)에서 구축한 sports_news_data와 합쳐 하나의 데이터셋을 만든다.
3) Summary (추출 요약문)을 문장 단위로 분리하였을 때 3문장이 되지 않거나, 3문장을 넘어가는 데이터의 경우, outlier로 간주하고 학습 데이터셋으로부터 제외한다.
4) 3)에서 얻은 최종 데이터셋 중 sports_news_data의 2,000건은 최종 평가용 test dataset으로 분리하고, 나머지 데이터만 학습 및 validation 평가에 사용한다.
5) 데이터셋을 MatchSum 모델의 input, 즉, (SENTENCE, DOCUMENT, BINARY_LABEL) (설명: 문장 1개, 기사 원문 전체, 문장 1개가 정답 요약문에 포함되어 있는지의 여부(0, 1))의 형태로 변환한다.
6) 모델에 데이터셋이 입력되면, sentence에 대한 embedding, document에 대한 embedding, 그리고 앞의 각각의 embedding을 element-wise product한 combined_features를 모두 concat한 형태를 모델이 input으로 받아 학습하고, 이 input이 classifier를 거쳐 최종적으로 0~1 사이의 확률값(특정 문장이 summary에 포함될 확률이 높을수록 숫자는 1에 가까워짐)을 return한다. 해당 확률값을 반올림하여 0 혹은 1로 만들면 predict 값이 되고, 이를 통해 정답(BINARY_LABEL)과의 평가를 진행한다.
7) 이렇게 학습된 모델을 사용하여 sports_news_data 총 9,075 건에 대한 inference를 진행하고, PREDICT SUMMARY(추출 요약문 3문장)를 도출하였다.

- **BertSum**[2]

[모델 정보]

- Pretrained Model: klue/bert-base
- Loss Function: BCELoss
- Optimizer: AdamW
- Learning Rate: 1e-5
- Epochs: 15
- Evaluation Metric: Rouge-1

[학습 과정]

1) sports_news_data.csv에 있는 기사 원문(CONTENT)에 대해 CLOVA Summary API를 사용하여 3문장으로 구성된 추출 요약문 정답셋을 구축한다.
2) AI Hub에서 추가적인 추출 요약문 데이터셋을 확보하고, 이를 1)에서 구축한 sports_news_data와 합쳐 하나의 데이터셋을 만든다.
3) Summary (추출 요약문)을 문장 단위로 분리하였을 때 3문장이 되지 않거나, 3문장을 넘어가는 데이터의 경우, outlier로 간주하고 학습 데이터셋으로부터 제외한다.

[2] https://github.com/nlpyang/PreSumm에서 참고

4) 3)에서 얻은 최종 데이터셋 중 sports_news_data의 2,000건은 최종 평가용 test dataset으로 분리하고, 나머지 데이터만 학습 및 validation 평가에 사용한다.
5) 데이터셋을 BertSum 모델의 input 형태로 변환한다. BertSum에서 필요로 하는 input 값들과 그에 대한 설명은 다음과 같다.

BertSum Input Features

1. src (document의 토큰들을 tokenizer를 거쳐 token id로 변환한 결과, list)
2. segs (summary 문장이 document 내에 어디에 위치하고 있는지 위치(position)를 알려주는 segment embedding, 0 OR 1로 이루어진 list)
3. mask_src (src에 대한 attention mask값, 0 OR 1로 이루어진 list)
4. clss (src 내에서 [CLS] 토큰이 위치하고 있는 string index 값, list)
5. mask_cls (summary 문장의 토큰 개수만큼 1이 채워진 리스트, 1로 이루어진 list)
6. src_sent_labels_list (document의 문장 중에서 summary 문장이 어디에 위치하고 있는지에 대한 문장 인덱스 정보, 0 OR 1로 이루어진 list)

6) Input feature를 넣어 생성된 embedding으로 모델 학습을 진행하고, 예측 결과로 나오는 sigmoid 확률값을 통해 cross entropy loss를 계산한다. 해당 loss를 줄여나가는 방향으로 weight을 학습한다.
7) 이렇게 학습된 모델을 사용하여 sports_news_data 총 9,075 건에 대한 inference를 진행하고, PREDICT SUMMARY(추출 요약문 3문장)를 도출하였다.

6. 모델 튜닝 과정

추출요약 모델의 구현, 학습 및 inference에 매우 오랜 시간이 소요되어, 모델의 hyperparameter tuning은 별도로 진행하지 않았다.

7. 최종 결과 분석

검증 데이터셋에 대한 모델의 최종 성능은 다음과 같았다.

Model	Rouge-1 Score
BertSum	0.7856

BertSum 모델은 뛰어난 추출 요약 성능을 보였다. MatchSum은 학습 및 추론 과정에서 엄청나게 오랜 시간이 걸린다는 단점이 있었다. 따라서 많은 양의 데이터를 학습 및 추론에 사용하기 위해서는 BertSum 모델을 사용하는 것이 더 효율적일 것이라는 생각을 했다.