11/13 GD01 프로젝트 보고서 by YSLEE

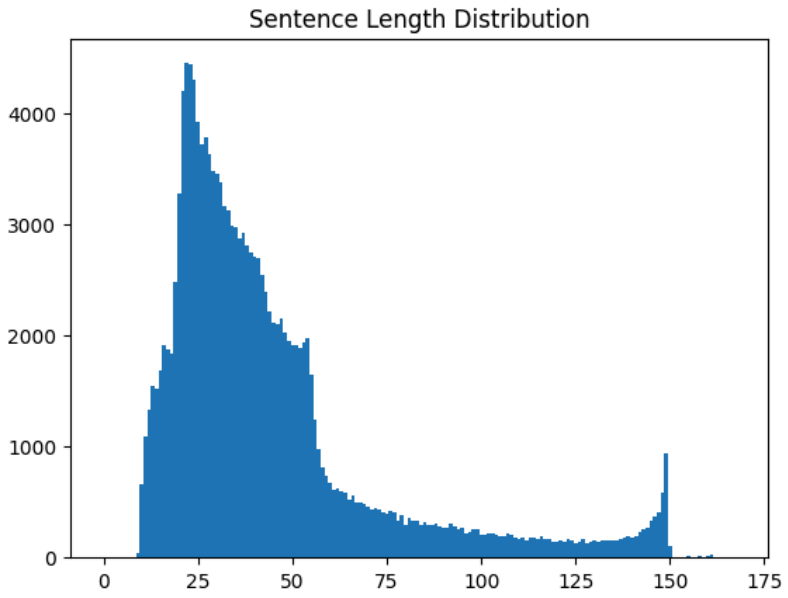
프로젝트 제출 루브릭

학습목표	평가기준
SentencePiece를 이용하여 모델을 만들기까지의 과정이 정상적으로 진행되었는가?	코퍼스 분석, 전처리, SentencePiece 적용, 토크나이저 구현 및 동작이 빠짐없이 진행되었는가?
SentencePiece를 통해 만든 Tokenizer가 자연어처리 모델과 결합하여 동작하는가?	SentencePiece 토크나이저가 적용된 Text Classifier 모델이 정상적으로 수렴하여 80% 이상의 test accuracy가 확인되었다.
SentencePiece의 성능을 다각도로 비교분석하였는가?	SentencePiece 토크나이저를 활용했을 때의 성능을 다른 토크나이저 혹은 SentencePiece의 다른 옵션의 경우와 비교하여 분석을 체계적으로 진행하였다.

1. 실험자료
    a. Base model: GD01_Base_mecab+sentencepiece_LSTMv2.ipynb
    b. 실험 #1: GD01_experiment01_okt+sentencepiece+LSTM.ipynb
    c. 실험 #2: GD01_experiment02_Mecabtoken+LSTM.ipynb
2. 코퍼스 분석
    a. ratings_train.txt

```
Data Size: 150001
Example:
>> id       document                                                label
>> 3989148  약탈자를 위한 변명, 이라. 저놈들은 착한놈들 절대 아닌걸요.   1
>> 4805788  이 영화가 왜 이렇게 저평가 받는지 모르겠다
1
>> 8317483  백봉기 언제나오나요?                                        1
>> 9801316  아햏햏 아햏햏 아햏햏.
1
```

문장의 최단 길이: 8
문장의 최장 길이: 168
문장의 평균 길이: 45


Sentence Length Distribution
4000
3000
2000
1000
0
0
25
50
75
100
125
150
175

3. 전처리 수행 항목
    a. 반복되는 문장부호(e.g. '…', ';;;') → 하나만 남기고 제거
    b. 반복 문자 제거
    c. 이모티콘, 특수문자 제거
    d. 맞춤법 변형 통일
        - 굩, 굳, 굿 ⇒ 굿
            1. 찾아보니까 "굳" 이게 "굳이"의 굳이 꽤 많아서 "굳이"를 제외한 "굳"만 바꿔야 할 것 같아요
            2. '굿'이 올바른 표현이나 예외 처리의 단순함을 위해 '굳'으로 진행.
        - ㅁㅊ, 미쳣, 미쳣다 → 미쳤
        - 괜찮, 괜춘, 괜찬, ㄱㅊ, 갠찬, 갠찮, 괸찬, 괸찮 → 괜찮
        - 봣 → 봤
        - 겟 → 겠
    e. 문장부호 앞뒤로 공백 추가
    f. 'ㅋ', 'ㅠ' 등 자음/모음 단독으로 존재 → 제거
    g. 영어 → 전부 소문자 처리
    h. stopwords **=**
    ['의','가','이','은','들','는','좀','잘','걍','과','도','를','으로','자','에','와','한','하다']
    불용어 제거
    i. 마지막에 연속 공백 하나로 통합
    j. 결측치, 중복 행 제거
    k. 전처리 결과 비교
        - 전처리 전 데이터 크기: 150000
        - 결측치 제거 후: 149995
        - 빈 문자열 제거 후: 149607
        - 중복 제거 후: 144478
            1. Train 데이터: 144478개
            2. Test 데이터: 48700개

4. 형태소 분석 결과 - 2가지 형태소 분석기가 완전히 다른 분석 결과를 보여줌.
    a. Mecab 형태소 분석기 결과

✅ 기본 통계:
- 분석 대상 문장: 5,000개
- 총 형태소 수: 79,694개
- 고유 형태소 수: 9,055개
- 평균 형태소/문장: 15.94개

✅ 품사 통계:
- 품사 종류: 217개
- 가장 많은 품사: NNG (19,854개)
- 가장 적은 품사: NNG+XSV+EC (1개)

✅ 형태소 통계:
- 최빈 형태소: . (4,539회)

- 한 번만 나온 형태소: 5,035개
- 평균 빈도: 8.80

📊 상위 15개 형태소 빈도

	형태소	빈도
0	.	4539
1	영화	1924
2	다	1801
3	고	1670
4	하	1381
5	이	1228
6	을	1018
7	는	916
8	보	815
9	,	744
10	게	734
11	지	647
12	없	584
13	들	574
14	있	531

📊 품사 분포

	품사	빈도	비율(%)
0	NNG	19854	24.91
1	EC	6331	7.94
2	MAG	5260	6.60
3	SF	4749	5.96
4	VV	4252	5.34
..	...	...	...
212	VV+EC+JX	1	0.00
213	ETM+NNB+JKG	1	0.00
214	NP+JKB+JX	1	0.00
215	XSV+EC+VX	1	0.00
216	NNG+XSV+EC	1	0.00

📊 형태소 길이 분포

	길이(글자)	빈도	비율(%)
0	1	42146	52.88
1	2	29896	37.51
2	3	6458	8.10

3	4	945	1.19
4	5	177	0.22
5	6	30	0.04
6	7	12	0.02
7	8	10	0.01
8	9	5	0.01
9	10	4	0.01
10	11	1	0.00
11	12	3	0.00
12	14	2	0.00
13	15	1	0.00
14	16	1	0.00
15	18	1	0.00
16	21	1	0.00
17	24	1	0.00

b. OKT 형태소 분석기 결과

✅ 기본 통계

- 분석 대상 문장: 5,000개
- 총 형태소 수: 64,154개
- 고유 형태소 수: 13,303개
- 평균 형태소/문장: 12.83개

✅ 품사 통계:

- 품사 종류: 17개
- 가장 많은 품사: Noun (30,344개)
- 가장 적은 품사: PreEomi (1개)

✅ 형태소 통계:

- 최빈 형태소: . (4,539회)
- 한 번만 나온 형태소: 8,388개
- 평균 빈도: 4.82

📊 상위 20개 형태소:

순위	형태소	빈도	비율(%)
1	.	4539	7.08%
2	영화	1699	2.65%
3	을	781	1.22%
4	,	744	1.16%
5	이	623	0.97%
6	!	512	0.80%
7	들	484	0.75%
8	?	478	0.75%
9	너무	333	0.52%

10	정말	332	0.52%
11	다	327	0.51%
12	만	323	0.50%
13	진짜	295	0.46%
14	점	283	0.44%
15	적	283	0.44%
16	에서	256	0.40%
17	로	247	0.39%
18	연기	225	0.35%
19	것	222	0.35%
20	에	221	0.34%

📊 품사 빈도 분포:

품사	빈도	비율(%)
Noun	30344	47.30%
Verb	8620	13.44%
Punctuation	6273	9.78%
Josa	6133	9.56%
Adjective	6042	9.42%
Suffix	1445	2.25%
Adverb	1429	2.23%
Modifier	1022	1.59%
Number	974	1.52%
Determiner	586	0.91%
Alpha	418	0.65%
VerbPrefix	370	0.58%
Exclamation	201	0.31%
Conjunction	148	0.23%
Foreign	121	0.19%
Eomi	27	0.04%
PreEomi	1	0.00%

📊 형태소 길이별 분포:

길이(글자)	빈도	비율(%)
1	22304	34.77%
2	27606	43.03%
3	9560	14.90%
4	3428	5.34%
5	1000	1.56%
6	194	0.30%
7	37	0.06%
8	11	0.02%
9	4	0.01%
10	4	0.01%

```
11          1           0.00%
12          2           0.00%
14          1           0.00%
15          1           0.00%
16          1           0.00%
==================================================
📈 형태소 길이 통계:
   평균: 1.97 글자
   최대: 16 글자
   최소: 1 글자
   중앙값: 2.00 글자
```

5. SentencePiece 적용 (max_length+130 설정)
    a. Base: Mecab+SentencePiece 적용
    b. 실험 #1: Okt+SentencePiece 적용
    c. 실험 #2: Mecab만 적용

6. 자연어처리 모델: LSTM
    a. 공통 하이퍼파라미터 정의
        - Vocab size = 8,000
        - EMBEDDING_DIM = 100
        - HIDDEN_DIM = 128
        - OUTPUT_DIM = 1
        - N_LAYERS = 2
        - BIDIRECTIONAL = True
        - DROPOUT_RATE = 0.2
        - PAD_IDX = 0

7. 실험 결과

실험 번호	Base	실험 #2	실험 #3
실험 요소	Mecab+SentencePiece+LSTM	Okt+SentencePiece+LSTM	Mecab+LSTM
실험 목적		Mecab과 Okt 형태소 분석기 성능 비교	SentencePiece 성능 분석
성능	Epoch 12 early stop Test Loss: 0.364 Test Acc: 84.42%	Epoch 11 early stop Test Loss: 0.373 Test Acc:	Epoch 11 early stop Test Loss: 1.021 Test Acc:

		83.72%	55.50%
소요시간	약 396초	약 350초	약 320초
목표 달성	달성	달성, 시간 소요 약 0.88	미달성
비교 발견사항	성능이 더 좋음	성능은 다소 떨어지나, 시간 소요가 더 적음	SentencePiece 의 활용이 성능을 크게 개선함 발견

8

8. 제약 및 next step
    a. 다른 npl 모델과 결합한 실험 추가 필요
    b. 타 형태소 분석기를 결합해서 실험하고자 하였으나, 셋팅 환경이나 가용성에서 제약이 많아서 실패함. 특히, 각 형태소 분석기마다 특성이 달라서 활용에 많은 애로사항 발생함. 이에 각각에 맞는 셋팅이 필요함을 알게 됨 (형태소 분석기 방랑기: Mecab, ETRI OPEN API, Kkma, Kiwi, KHAIII -> 엄청난 실패 후 Mecab과 OKT 만 성공적으로 실험