자연어처리를 활용하여 감정분석하기
(Sentiment Analysis)

인공지능학부 20223201 임혜진

1. 데이터 전처리

자연어처리를 활용한 감정 분석 과제를 수행하기 위해 Kaggle에서 제공하는 트위터 감정 분석 데이터셋을 사용했습니다. 이 데이터셋은 다음과 같은 10개의 컬럼으로 이루어져 있습니다.

['textID', 'text', 'selected_text', 'sentiment', 'Time of Tweet', 'Age of User', 'Country', 'Population -2020', 'Land Area (Km²)', 'Density (P/Km²)]

이 중에서 제가 감정 분석에 사용한 주요 컬럼은 text와 sentiment입니다. text는 트윗의 내용이고, sentiment는 각 text마다의 감정 분류입니다. neutral, negative, positive 3가지 label로 구성됩니다.

text와 sentiment만으로 데이터프레임을 재정의하면 다음과 같습니다.

	text	sentiment
0	I`d have responded, if I were going	neutral
1	Sooo SAD I will miss you here in San Diego!!!	negative
2	my boss is bullying me...	negative
3	what interview! leave me alone	negative
4	Sons of ****, why couldn`t they put them on t...	negative
...	...	...
27476	wish we could come see u on Denver husband l...	negative
27477	I`ve wondered about rake to. The client has ...	negative
27478	Yay good for both of you. Enjoy the break - y...	positive
27479	But it was worth it ****.	positive
27480	All this flirting going on - The ATG smiles...	neutral

27481 rows × 2 columns

데이터셋 전처리 과정에서 수행한 작업은 다음과 같습니다.

1-1. 결측값(NA) 제거

먼저 df.isna().sum()을 통해서 데이터 프레임의 행별 결측값을 확인해보았습니다. 출력 결과 sentiment에는 결측값이 없었고, text에서 1개의 결측값이 존재했습니다. df.dropna(inplace=True) 를 통해 결측값이 있는 행을 삭제하였습니다.

1-2. text 정규화

text의 일관성을 높이고 깔끔하게 만들기 위해서 text의 정규화 작업을 진행했습니다. 정규화 작업 내용은 아래와 같습니다.

- text의 모든 문자를 소문자로 변환하여 대소문자를 통일
- 단어와 공백을 제외한 모든 문자(문장 부호나 특수 문자)를 제거
- 중복된 공백을 하나로 교체하고, 양끝에 공백이 있을 경우 제거

1-3. 불용어(stop words) 제거

I, me, the 같은 조사나 접미사, 몇몇 대명사의 경우 자주 등장하지만 분석을 하는 것에 있어서는 큰 도움이 되지 않습니다. 따라서 이러한 불용어를 제거해야합니다. nltk가 정의한 영어 불용어를 다운받고, remove_stopwords() 함수를 통해 불용어를 제거했습니다.

1-4. sentiment별로 label을 생성

neutral에 0, negative에 1, positive에 2로 label을 붙여주었습니다. label은 데이터프레임의 열로 추가했습니다.

전처리 과정을 수행하면서의 데이터프레임 변화는 아래와 같습니다.

	text	sentiment
0	I`d have responded, if I were going	neutral
1	Sooo SAD I will miss you here in San Diego!!!	negative
2	my boss is bullying me...	negative
3	what interview! leave me alone	negative
4	Sons of ****, why couldn`t they put them on t...	negative
...	...	...
27476	wish we could come see u on Denver husband l...	negative
27477	I`ve wondered about rake to. The client has ...	negative
27478	Yay good for both of you. Enjoy the break - y...	positive
27479	But it was worth it ****.	positive
27480	All this flirting going on - The ATG smiles...	neutral

27480 rows × 2 columns

1. 필요한 행만 추출, 결측값 제거

	text	sentiment
0	id have responded if i were going	neutral
1	sooo sad i will miss you here in san diego	negative
2	my boss is bullying me	negative
3	what interview leave me alone	negative
4	sons of why couldnt they put them on the relea...	negative
...	...	...
27476	wish we could come see u on denver husband los...	negative
27477	ive wondered about rake to the client has made...	negative
27478	yay good for both of you enjoy the break you p...	positive
27479	but it was worth it	positive
27480	all this flirting going on the atg smiles yay ...	neutral

27480 rows × 2 columns

2. text 정규화

	text	sentiment
0	id responded going	neutral
1	sooo sad miss san diego	negative
2	boss bullying	negative
3	interview leave alone	negative
4	sons couldnt put releases already bought	negative
...	...	...
27476	wish could come see u denver husband lost job ...	negative
27477	ive wondered rake client made clear net dont f...	negative
27478	yay good enjoy break probably need hectic week...	positive
27479	worth	positive
27480	flirting going atg smiles yay hugs	neutral

27480 rows × 2 columns

3. 불용어 제거

	text	sentiment	label
0	id responded going	neutral	0
1	sooo sad miss san diego	negative	1
2	boss bullying	negative	1
3	interview leave alone	negative	1
4	sons couldnt put releases already bought	negative	1
...	...	...	...
27476	wish could come see u denver husband lost job ...	negative	1
27477	ive wondered rake client made clear net dont f...	negative	1
27478	yay good enjoy break probably need hectic week...	positive	2
27479	worth	positive	2
27480	flirting going atg smiles yay hugs	neutral	0

27480 rows × 3 columns

4. label 생성

테스트 데이터에 대해서도 위와 동일하게 전처리 과정을 진행했습니다.

2. 모델 설계

모든 자연어 처리 분야에서 좋은 성능을 내고 있는 BERT 모델을 이용하여 감정 분석을 진행했습니다. BERT는 트랜스포머의 인코더 블록을 적용한 모델입니다. 모든 레이어에서 양방향 문맥 특성을 활용할 수 있기에 감정 분석에 적합하다고 판단하였습니다. 또한 대용량 말뭉치로 사전학습이 진행되어 있는 모델이기에 언어에 대한 이해도가 높습니다.

감정 분류 모델로는 'BertForSequenceClassification'의 bert-base-uncased 모델을 사용했습니다. 모델은 입력 임베딩(Embeddings), 인코더(Encoder), 풀러(Pooler), 분류 헤드(Classifier)로 이루어져 있습니다. 모델의 학습 매개변수를 업데이트하기 위한 Optimizer는 AdamW를 사용하였고, 학습률을 조정하기 위해서 Learning Rate Scheduler를 설정하였습니다. 배치 사이즈를 32로 하여 dataloader를 생성하고 모델 학습을 진행하였습니다.

3. 결과 분석

과제를 진행해보면서 점점 더 방향성을 잡을 수 있었습니다. 따라서 코드 버전을 바꾸면서 기존 코드를 기반으로 다른 시도를 추가해보았습니다.

Version 1

첫 번째 버전에서는 결측값 삭제 정도로만 데이터 전처리를 진행하고 학습 및 평가를 진행했습니다. 그 결과 테스트 정확도는 약 0.40이 나왔습니다. 성능이 낮게 나온 이유가 불충분한 데이터 전처리였음을 알고 두 번째 버전으로 전처리를 더 진행했습니다.

Version 2

두 번째 버전에서는 결측값 삭제와 더불어 텍스트 정규화, 불용어 제거까지 전처리를 진행했습니다. 훨씬 깔끔하고 정제된 text로 학습을 진행한 결과, 테스트 정확도는 0.75가 나왔습니다. 이전 버전 대비 0.35나 높아진 수치로 훨씬 성능이 좋아진 것을 확인할 수 있었습니다. 클래스 별 정확도는 아래와 같습니다.

Ver 2. result	Accuracy
neutral	1026/1430 (71.75%)
negative	753/1001 (75.22%)
positive	883/1103 (80.05%)
overall	2662/3534 (**75.33**%)

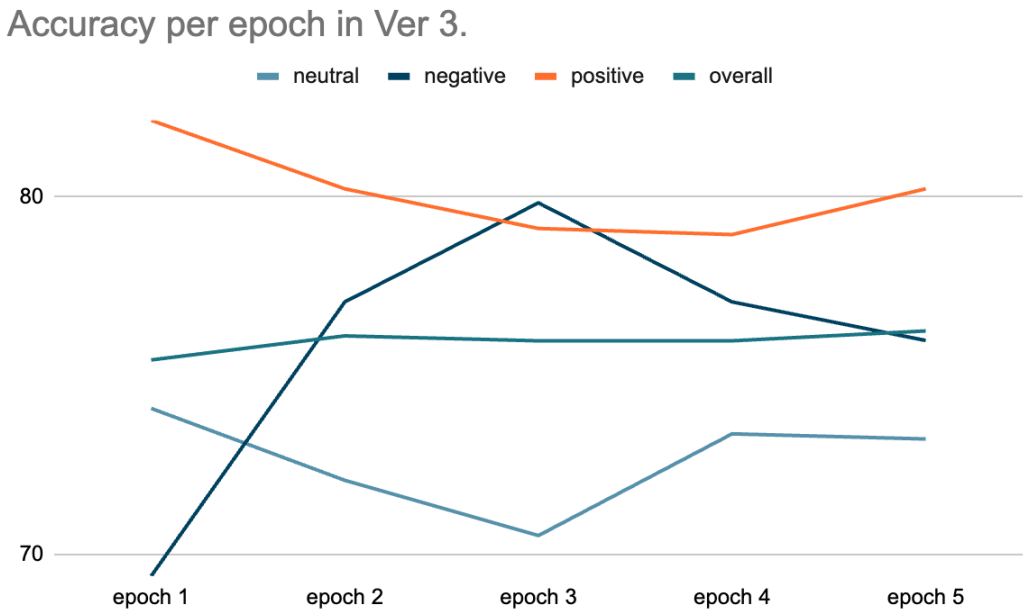
Version 3

세 번째 버전에서는 train.csv의 전체 데이터를 모두 학습에 사용해보았습니다. 첫 번째와 두 번째 버전에서는 train.csv 데이터를 training과 validation 데이터로 쪼개는 작업을 진행 후 검증까지 하였는데, test.csv데이터가 주어지므로 굳이 검증을 하지 않아도 될 것 같다고 판단했습니다. 따라서 split하는 대신 train.csv의 전체 데이터를 학습에 사용하였고 test.csv 데이터로 테스트를 했습니다. 세 번째 버전의 테스트 정확도는 아래와 같습니다. 더 많은 양의 데이터를 학습 과정에 넣어주었으나 성능이 크게 증가하지는 못했습니다.

Ver 3. result	Accuracy
neutral	1045/1430 (73.08%)
negative	759/1001 (75.82%)
positive	885/1103 (80.24%)
overall	2689/3534 (**76.09**%)

세 번째 버전에서는 데이터 외에도 하이퍼파라미터인 에폭을 수정했습니다. 기존에 3이었던 에폭을 5로 늘려 더 충분히 데이터를 학습할 수 있도록 하였습니다. 그 결과 에폭이 증가할수록 loss가 감소하는 것을 확인할 수 있었습니다. 에폭이 증가할수록 테스트 결과 정확도도 향상하는 것을 기대했습니다. 그러나 에폭 별로 저장된 모델들을 모두 테스트 시켜본 결과 에폭 1~5 사이의 성능 차이는 크게 없었습니다.

아래는 에폭에 따른 Accuracy 결과를 표와 선 차트로 나타낸 것입니다.



	epoch 1	epoch 2	epoch 3	epoch 4	epoch 5
neutral	1057/1430 (73.92%)	1029/1430 (71.96%)	1008/1430 (70.49%)	1047/1430 (73.22%)	1045/1430 (**73.08**%)
negative	695/1001 (69.43%)	770/1001 (76.92%)	799/1001 (79.82%)	770/1001 (76.92%)	759/1001 (**75.82**%)
positive	908/1103 (82.32%)	885/1103 (80.24%)	872/1103 (79.06%)	870/1103 (78.88%)	885/1103 (**80.24**%)
overall	2660/3534 (75.27%)	2684/3534 (75.95%)	2679/3534 (75.81%)	2687/3534 (76.03%)	2689/3534 (**76.09**%)

4. 느낀 점

한 학기 동안 자연어처리 과목을 수강하면서 많은 것을 배웠습니다. 수강신청을 할 때는 자연어처리에 대해 아무것도 알지 못했는데 word embedding부터 GPT까지 차근차근 배워나간 것 같습니다. BERT와 GPT처럼 요즘 많이 쓰이는 최신 기술들에 대해 배울 때는 이론적인 공부에서 그치지 않고 실제로 코드를 작성하면서 모델을 사용해보고 싶었는데, 마침 과제로 주어져서 좋은 기회라고 생각하고 과제에 임했습니다. 수업 시간에 배웠던 것들을 실제로 사용해보니 더 잘 이해가 되었습니다. 버전을 변경해보면서 성능 향상 시켜보는 과정에서는 데이터 전처리 과정이 중요함을 느낄 수 있었습니다. 또 코드를 개선시킬 수록 성능이 향상되는 것을 보면서 뿌듯했고 어떻게 해야 더 성능을 올릴 수 있을까 스스로 고민하는 시간을 가질 수 있었던 것 같습니다. 과제로는 BERT만 사용해서 감정분석을 진행했지만 GPT나 LSTM 등 다른 모델들도 사용해서 모델들간의 성능도 비교해보고 싶습니다.