기계학습 및 데이터마이닝
헤이 런닝머신, 이 노래 반응이 어때?

(부제 : SoundCloud Music Rating Summarizing Service)

교수님 : 손경아 교수님
제출 일 : 2020.06.17
조 이름 : 런닝머신

이름	학번
임윤진	201723825
고예준	201820742
박민석	201820793
배해진	201820730
박효성	201421092

아주대학교
AJOU UNIVERSITY

1.Business Understanding

1. Background

SoundCloud(https://soundcloud.com/)는 개인이 자유롭게 음반을 제작하고 음원을 공유할 수 있는 온라인 플랫폼이다. 기존의 SoundCloud 는 특정 음반에 대한 개개인의 평점이 표시되지 않고 댓글만 남길 수 있었다. 그래서 다른 사람들이 특정 음악에 대해 어떻게 생각하는지를 간단히 파악하기 어렵고, 댓글들을 모두 읽어 봐야한다는 단점이 있었다. 요약되고 함축된 것으로 쉽게 의미를 파악하는 것을 선호하는 현대인이 많은 만큼 다른 사람들이 특정 음반에 대해서 어떻게 생각하는지 빠르게 알고 싶어 하는 니즈가 있을 것이라고 판단했다.

2. Describe all requirements, assumptions, risks and constraints

[requirements]
이진 분류 모델 생성 시, 긍정적인 리뷰(overall : 4, 5 점)가 부정적인 리뷰(overall :1, 2 점) 대비 상당히 많이 존재하므로, 이들의 데이터의 비율을 맞춰주는 것이 요구된다.
요약 모델 생성 시, 리뷰에 쓰인 단어들로 단어 사전을 만들 때, 자주 쓰이지 않는 희귀한 단어들에 대해 처리하는 과정이 요구된다.
[assumptions]
아마존 리뷰데이터에서 사용자들이 남겼던 리뷰의 긍정문, 부정문의 댓글들과 SoundCloud 사용자의 댓글의 긍정문, 부정문 댓글들 중 긍정, 부정을 의미하는 단어가 비슷할 것이라고 가정을 하고 시작하였다.
[risks]
아마존 리뷰데이터는 아마존에서 판매하는 음원에 대해서 음원 구매자가 리뷰를 올리는 것이고 SoundCloud 는 SoundCloud 에서 단순히 올리는 음원에 대한 단순 comment 이기 때문에 두 개의 데이터의 출처가 다르다고 볼 수 있다. 따라서 두 데이터로부터 생성되는 단어 사전이 다를 것이고, 아마존 리뷰데이터에는 없는 단어에 대하여 결측값 발생을 염두 할 수 있다.
[constraints]
아마존 리뷰데이터의 경우, 아마존에서 제공하는 데이터를 통해 리뷰 데이터를 얻을 수 있었지만, SoundCloud 의 경우, SoundCloud 에서 제공하는 API, 데이터가 없으므로 직접 모든 음악들에 관한 댓글들을 크롤링을 해야한다.

3. Determine data mining goals and success criteria

학습된 이진 분류 모델과 요약 모델을 통해서 SoundCloud 의 댓글을 긍정과 부정으로 각각 분류한 후, 긍정에 대한 요약문과 부정에 대한 요약문을 도출할 것이다. 요약

모델에 대한 성능은 'ROUGE-N'이라는 성능 평가 척도를 통해 평가할 수 있다. 기존 요약 모델 연구에서의 모델 성능을 고려했을 때, ROGUE-1 는 약 20~44% 이상, ROGUE-2 는 약 10~21% 이상의 값을 기대한다.

4. Project plan

	일자	설명
1	20.05.15	프로젝트 제안서 작성
2	20.05.20	프로젝트 제안서 수정
3	20.06.03	아마존 데이터 전처리, soundCloud 데이터 크롤링
4	20.06.09	SoundCloud 데이터 전처리, 이진 분류 모델 생성, 요약 모델 생성
5	20.06.16	이진 분류 모델 평가, 요약 모델 평가, 최종 보고서 작성
6	20.06.17	최종 보고서 작성 및 프로젝트 ppt 완성

2.Data Understanding

1. Describe the data you use or how you collect the data

Digital Music reviews (1,584,082 reviews)

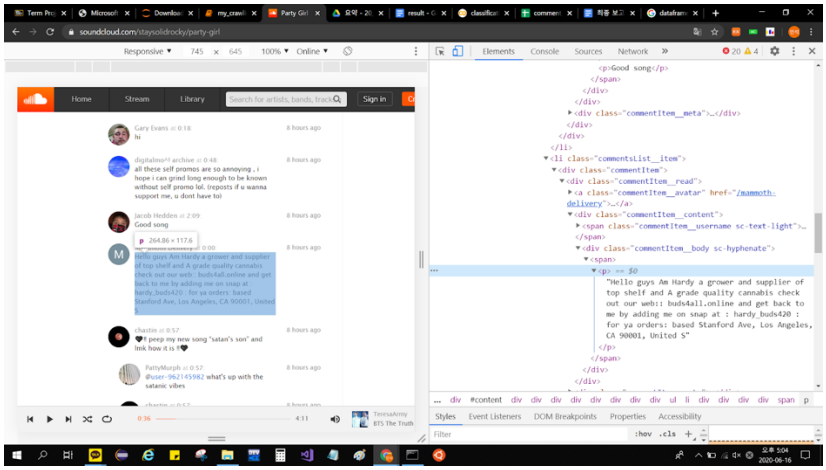

```
{
"image": ["https://images-na.ssl-images-
amazon.com/images/I/71eG75FTJJL._SY88.jpg"],
"overall": 5.0,
"vote": "2",
"verified": True,
"reviewTime": "01 1, 2018",
"reviewerID": "AUI6WTTT0QZYS",
"asin": "5120053084",
"style": {
        "Size:": "Large",
        "Color:": "Charcoal"
        },
"reviewerName": "Abbey",
"reviewText": "I now have 4 of the 5 available colors of this
shirt... ",
"summary": "Comfy, flattering, discreet--highly recommended!",
"unixReviewTime": 1514764800
}
```


Figure 1

	comment
0	\n❤ CHRISTIAN MUSIC SIMILAR-SOUNDING TO THIS A...
1	\nFire ass song on my page if it isn't fire I'...
2	\n🔥🔥Yoo im 14 and dropped 2 actually fire sing...
3	\n@y_mil_lions 😁YESSIRRRR. 😏□ im a slowly risi...
4	\n@user-247278830 😁YESSIRRRR. 😏□ im a slowly r...
...	...
4245	\nI'm a 17 year old rapper from Chicago and yo...
4246	\nis this tik tok\n
4247	\nEveryone's saying I got UK Post Malone/ Juic...
4248	\nAll beats $7.99 or under for a basic license...
4249	\nparty girl!!\n

Figure 2

이번 프로젝트에서 사용한 데이터는 아마존의 제품 리뷰데이터 중 Digital Music data 를 선택하여 수집한 데이터이다. 1,584,082 개 중 60,000 개를 사용하였다. 아마존 데이터는 figure1 처럼 overall, vote, verified 등의 여러가지 feature 로 구성되어 있지만, 이번 모델 생성을 위해 overall, reviewtext, summary feature 를 사용하였다. Overall 은 사용자가 음원에 대해서 평가한 점수(1 점-5 점)이고, reviewtext 는 리뷰 전문에 관한 데이터이고, summary 는 리뷰의 요약을 나타낸다.
Figure2 은 SoundCloud 에 있는 음악 중 'party girl'이라는 음악에 대한 댓글을 크롤링한 4250 개의 데이터이다.

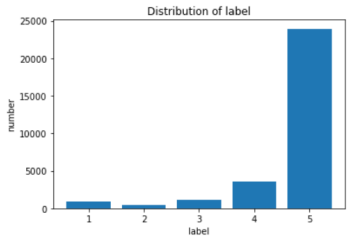


SoundCloud 의 댓글을 크롤링하기 위해 웹 페이지 소스 보기(F12)를 통해 사이트의 HTML 이 어떤 구조를 가지고 있는지 살펴본 후, 셀레니움, 뷰티풀 숲을 사용하여 댓글이 들어있는 태그를 추출하여 따로 저장했다.

2.Exploratory Data Analysis

	overall	reviewText	summary	pretreatment
0	5.0	this is a great cd full of worship favorites!!...	great worship cd	great cd full worship favorite time great ke...
1	5.0	so creative! love his music - the words, the ...	gotta listen to this!	creative love music word message favorite ...
2	5.0	keith green, gone far to early in his carreer,...	great approach still gets the message out	keith green gone far early carreer left u go...
3	5.0	keith green had his special comedy style of ch...	great a must have	keith green special comedy style chirstian mus...
4	5.0	keith green / so you wanna go back to egypt.....	a great one from keith with a guest appearance...	keith green wan na go back egypt : album kei...
...	...	...	...	...
29995	5.0	soundgarden is an excellent group. they have o...	song	soundgarden excellent group one sound unique b...
29996	4.0	black hole sun is a terrific, unique song. as ...	black hole sun is a terrific, unique song	black hole sun terrific unique song oldie add...
29997	5.0	this was one of my favorite songs as well as f...	black hole sun	wa one favorite song well favorite video sound...
29998	4.0	very funny and catchy	four stars	funny catchy
29999	5.0	just like the radio.	five stars	like radio

30000 rows × 4 columns



아마존 디지털 음악 리뷰 데이터에서 평점의 분포를 히스토그램으로 그려본 결과, 평점이 5 인 리뷰가 월등히 많고, 1 점과 2 점인 리뷰는 상대적으로 적음을 알 수 있었다.

3. Data quality reports

아마존 음원 리뷰 데이터는 기본적으로 음원을 구매한 사람만 남긴 리뷰였기 때문에 비교적으로 잘 정돈된 데이터 형식을 보였다.
SoundCloud 댓글 데이터는 스팸 메시지 뿐만 아니라 동일한 댓글들이 많이 보였다. 또한 광고성 댓글들 또한 많아서 필터링이 필요하였다.

3.Data Preparation

1. Data cleaning / pre-processing (Review(댓글) 데이터 전처리)

	reviewText	pretreatment
0	this is a great cd full of worship favorites!!...	great cd full worship favorite time great ke...
1	so creative! love his music - the words, the ...	creative love music word message favorite ...
2	keith green, gone far to early in his carreer,...	keith green gone far early carreer left u go...
3	keith green had his special comedy style of ch...	keith green special comedy style chirstian mus...
4	keith green / so you wanna go back to egypt.....	keith green wan na go back egypt : album kei...
...	...	...
59995	just what i was looking for! needed a couple o...	wa looking needed couple tune cd complete com...
59996	a perfect match, old skool music with lyrics t...	perfect match old skool music lyric represent...
59997	great version	great version
59998	one of my favorties	one favorties
59999	great song	great song

60000 rows × 2 columns

Figure 3

1	\nFire ass song on my page if it isn't fire I'...	fire song page nt fire ll pay 5 paypal real s...
2	\n🔥🔥Yoo im 14 and dropped 2 actually fire sing...	yoo im 14 dropped 2 actually fire single come ...
3	\n@y_mil_lions 😡YESSIRRRR 😡□ im a slowly risi...	y_mil_lions yessirrrr im slowly rising artis...
4	\n@user-247278830 😡YESSIRRRR 😡□ im a slowly r...	user247278830 yessirrrr im slowly rising art...
6	\n@user-247278830\n	user247278830
...	...	...
4239	\nCheck out This 14 year old rapper's new sing...	check 14 year old rapper s new single agartha ...
4241	\nDeft\n	deft
4242	\n@kxngrjj\n	kxngrjj
4243	\ngo follow me on Instagram its 0sexygotyourlo...	go follow instagram 0sexygotyourlove0
4247	\nEveryone's saying I got UK Post Malone/ Juic...	everyone s saying got uk post malone juice wrl...

1862 rows × 2 columns

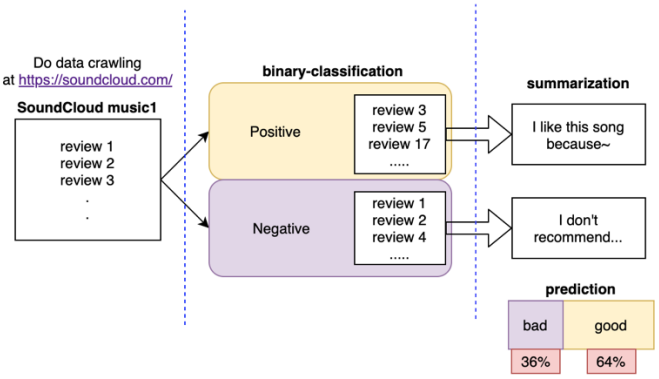
Figure 4

2. Describe in details if you construct the new data

음악데이터에서 사용되는 댓글 부분을 소문자로 변경을 진행하였다.
표제어를 추출하고 불용어 제거(nltk 모듈 사용, 특수문자) 와 댓글에 불필요한 부분인 이모티콘들을 제거 하였다. 또한 원문 중 중복되는 글, null 값을 제거 하였다.
단순 comment 만을 남겨놓는 방식으로 전처리를 완료 하였다.
위의 figure 3 은 아마존 음원 데이터 리뷰텍스트에 관한 전처리 작업 전 후이다. 위의 figure 4 은 SoundCloud 댓글 전처리 작업 전 후이다.

4. Modeling

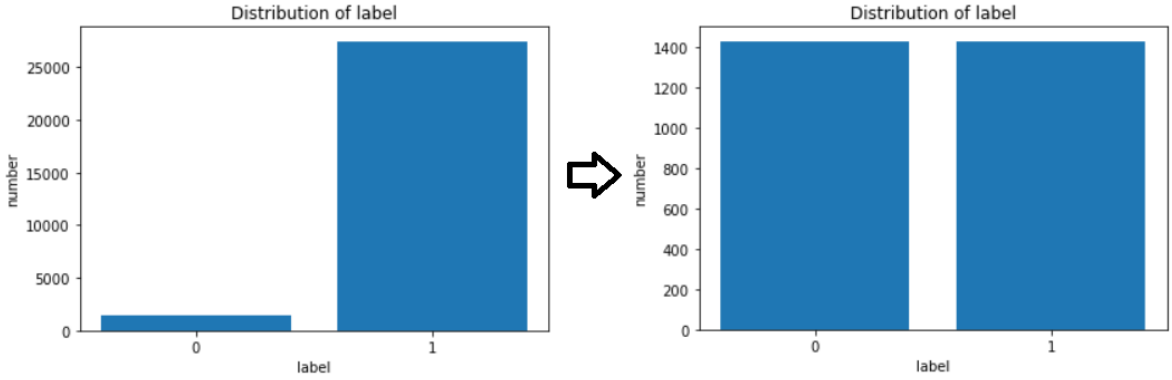
1. Model / Algorithm description



설계한 모델은 다음과 같이 SoundCloud 에 업로드 된 한 음악의 댓글들을 인풋으로 넣고, 학습된 이진 분류 모델로 긍정/ 부정을 분류한다. 그 다음 모델이 예측한 클래스의 댓글들을 각각 한 문장으로 요약을 한다. 또한, 긍정/ 부정 클래스에 속해있는 댓글의 개수와 비율을 계산하여 한 음악에 대한 전반적인 평가를 알 수 있도록 진행하였다.

2. 이진분류

전처리된 아마존 dataset 을 이용하여 이진분류 하였다. 평점 즉 overall 이 5.0, 4.0 인 것은 긍정으로 분류해 overall 을 1 로 바꾸고, overall 이 2.0, 1.0 인 것은 부정으로 분류해 overall 을 0 으로 바꾸었다.



데이터 개수를 분석해보니 긍정데이터의 개수는 2748 개, 부정데이터의 개수는 1432 개였다. 이때 성능 평가의 신뢰도를 높이기 위해, 긍정, 부정의 데이터개수를 각각 1432 개로 맞추어 주었다.

1) 전처리된 text 의 vector 화

1
```
['music trash original versions buy   listen music first wo nt like ',
 'country make music bad get private chopped off thats say  50 cent  nt birthday  re gon na party like birthday  especially crappy mu
sic nice day',
 'album show far hiphop ha gone down remember day eric b rakim good hiphop used be cd glorifies violence  womenizing negative things
ca nt teaching child  need positive music uplift culture ',
 'ok ill admit half world started listenin 50 wa listenin old cd s like  guess whose back  power dollar  cool wa tryin rap wa decent
8mile spit song  place go  thought wa gettin better rushed go comp album really listened like first 2pac album said :  man sell  clea
```

2
```
[[6, 525, 34, 24, 23, 6, 37, 315, 3, 7], [231, 21, 6, 50, 16, 924, 604, 32, 18, 41, 3, 1111, 93, 539, 241, 605, 7, 1111, 352, 1513,
6, 142, 80], [5, 169, 205, 353, 13, 768, 1514, 260, 80, 1269, 540, 1270, 11, 353, 196, 1185, 8, 806, 1050, 67, 3, 559, 104, 891, 6, 1
271], [157, 1515, 855, 385, 143, 372, 18, 4, 71, 8, 2, 7, 345, 1516, 43, 526, 674, 179, 4, 675, 35, 4, 325, 1, 292, 46, 91, 4, 48, 4
6, 5, 20, 211, 7, 37, 527, 5, 163, 144, 346, 560, 856, 1051, 6, 807, 237, 60, 35, 769, 8, 82, 1391, 70, 71, 102, 18, 420, 35, 232, 54
1, 857, 1392, 410, 157, 71, 269, 76, 1112, 73, 5, 3, 46, 24, 23, 60, 35, 7, 527, 1052, 1393, 35, 1651, 373, 507, 640, 347, 892, 219,
1517, 1272], [575, 16, 72, 43, 303, 87, 1273, 925, 223, 223, 223, 223], [270, 45, 282, 770, 180, 311, 431, 395, 132], [12, 18, 41, 1
```

3
```
[[   6  525   34 ...    0    0    0]
 [ 231   21    6 ...    0    0    0]
 [   5  169  205 ...    0    0    0]
 ...
 [ 485  350 1725 ...    3  620   97]
 [  10    8  499 ...    0    0    0]
 [  10    1   24 ...    0    0    0]]
```

① 전처리된 text 의 list 를 만든다.
② keras.preprocessing.text 의 Tokenizer 를 생성한후 생성된 Tokenizer 의 fit_on_texts 를 이용하여 단어사전을 만든다. 이때 총 단어집합의 크기는 11519 개이지만, 등장빈도가 6 이하인 희귀단어가 9701 가 있어 이를 제외한 1818 개를 이용하여 단어사전을 생성하였다. 단어사전 중 빈도수가 높은 단어를 사용하여 전처리된 text 를 vector 화 시켜준다.

```
단어 집합(vocabulary)의 크기 : 11519
등장 빈도가 6번 이하인 희귀 단어의 수: 9701
단어 집합에서 희귀 단어를 제외시킬 경우의 단어 집합의 크기 1818
단어 집합에서 희귀 단어의 비율: 84.21737998090111
전체 등장 빈도에서 희귀 단어 등장 빈도 비율: 19.12320314770517
```

③ 가장 긴 단어사전의 길이에 맞추어 zero padding 을 해준다

2) 모델

Train 할 모델로 keras 의 LSTM 을 사용하였다. Sequential 한 모델을 생성하여 1 개의 layer 를 가지고 있는 LSTM 을 추가하였다. 모델 평가 방법은 Cross Validation 을 이용하였고, 이때 validation loss 가 어느정도 줄어들다가 어느 순간 다시 커지는 상황에서 overfitting 을 막기 위해 학습을 멈출 수 있는 parameter 인 patience 는 1 로 설정하였다.

```
y_model = Sequential()
y_model.add(Embedding(1818, 120))
y_model.add(LSTM(1,activation='linear'))
y_model.add(Dense(2, activation='softmax'))
```

```
y_model = Sequential()
y_model.add(Embedding(1818, 120))
y_model.add(LSTM(1,activation='tanh'))
y_model.add(Dense(2, activation='softmax'))
```

```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=1)
mc = ModelCheckpoint('you2_model.h5', monitor='val_acc', mode='max', verbose=1, save_best_only=True)
```

테스트 정확도: 0.6167

테스트 정확도: 0.8571

```
y_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
```

이때 이진 분류를 위해 softmax 를 사용하여 확률 계산을 하였고, 'binary_crossentropy'를 통해 확률이 더 높은 label 을 선택하였다.

LSTM 의 activation function 을 Linear 로 하였을 경우, 테스트 정확도는 0.6167 이지만 activation function 을 Tanh 로 설정하였을 경우 테스트의 정확도는 0.8571 로 더 높게 나왔다. 따라서 activation function 은 Tanh 로 결정하였다.

```
y_model = Sequential()
y_model.add(Embedding(1818, 120))
y_model.add(LSTM(50,activation='tanh'))
y_model.add(Dense(2, activation='softmax'))
```

```
y_model = Sequential()
y_model.add(Embedding(1818, 120))
y_model.add(LSTM(100,activation='tanh'))
y_model.add(Dense(2, activation='softmax'))
```

```
y_model = Sequential()
y_model.add(Embedding(1818, 120))
y_model.add(LSTM(1818,activation='tanh'))
y_model.add(Dense(2, activation='softmax'))
```

```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=1)
mc = ModelCheckpoint('you2_model.h5', monitor='val_acc', mode='max', verbose=1, save_best_only=True)
```

테스트 정확도: 0.8641

테스트 정확도: 0.8920

테스트 정확도: 0.6167

LSTM 의 unit 개수를 바꾸어 보았다. 이때 첫번째 결과에 의해 unit = 1 -> 0.8571, 위의 결과에 의해 unit = 50 -> 0.8641, unit = 100 -> 0.8920, unit = 1818 -> 0.6167 이 도출되었다. Unit 이 너무 큰 경우만 아니면 테스트 정확도는 비슷하게 나온다.

```
y_model = Sequential()
y_model.add(Embedding(1818, 120))
y_model.add(LSTM(100,activation='tanh'))
y_model.add(Dense(2, activation='softmax'))
```

```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=2)
mc = ModelCheckpoint('you2_model.h5', monitor='val_acc', mode='max', verbose=1, save_best_only=True)
```
테스트 정확도: 0.8676

```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=3)
mc = ModelCheckpoint('you2_model.h5', monitor='val_acc', mode='max', verbose=1, save_best_only=True)
```
테스트 정확도: 0.8955

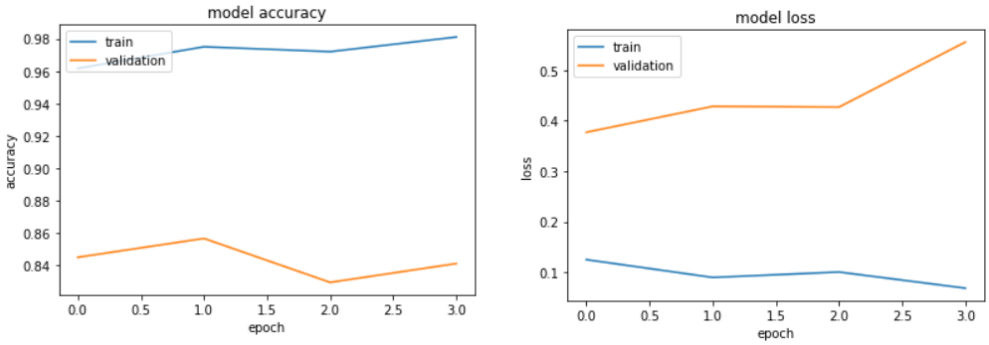
```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=4)
mc = ModelCheckpoint('you2_model.h5', monitor='val_acc', mode='max', verbose=1, save_best_only=True)
```
테스트 정확도: 0.8815

Cross Validation 를 결정하는 patience 의 parameter 를 바꾸어 보았다 먼저 patience = 1 일경우는 위의 결과에 의하여 0.8920, patience = 2 일 경우 0.8676, patience = 3 일 경우 0.8955, patience = 4 일 경우 0.8815 이다.

```
y_model = Sequential()
y_model.add(Embedding(1818, 120))
y_model.add(LSTM(100,activation='tanh'))
y_model.add(Dense(2, activation='softmax'))
```

```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=3)
mc = ModelCheckpoint('you2_model.h5', monitor='val_acc', mode='max', verbose=1, save_best_only=True)
```

⇨ 테스트 정확도: 0.8955



앞의 과정을 통하여, 얻은 것 중 가장 성능이 좋았던 파라미터들을 지정하여 모델을 학습하였다. 그 결과, 0.8955 라는 정확도를 얻을 수 있었다. 위의 그래프를 통해, model 의 accuracy 는 높아지면서 model 의 loss 는 작아지는 형태임을 알 수 있었다.

3. 요약 모델

1) Seq2Seq 모델을 기반으로, Attention Mechanism 을 도입한다.

Attention 은 디코더(decoder)에서 출력 단어를 예측하는 매 시점(time step)마다, 인코더(encoder)에서의 전체 입력 문장을 다시 한 번 참고한다. 이때 해당 시점에서 예측해야 할 단어와 연관이 있는 입력 단어 부분을 좀 더 집중(attention)해서 보게 된다.

2) 텍스트 제한길이 하이퍼파라미터

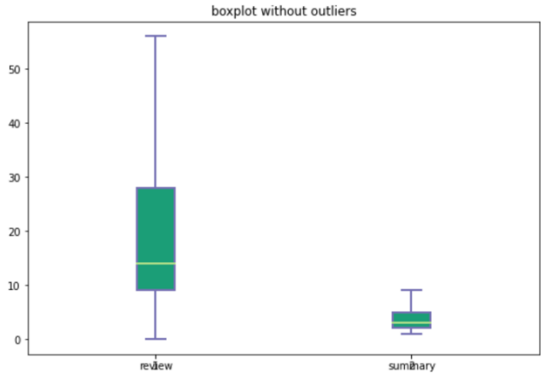
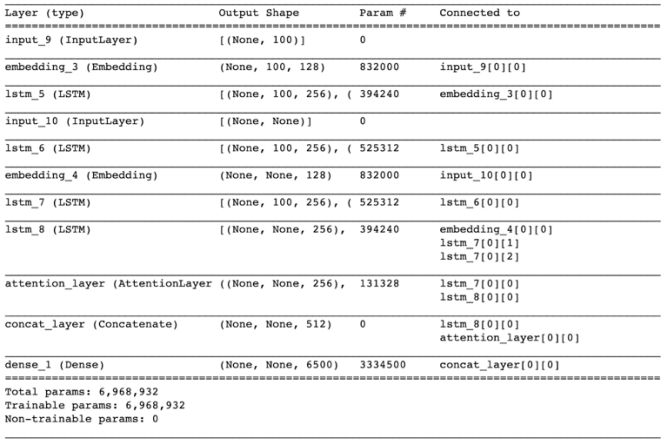


Figure 5

Figure 5 는 리뷰 데이터에서 중복 데이터를 제거하고, 각 문장의 길이를 측정한 후, outlier 를 제외하고 그린 boxplot 이다. 리뷰 데이터의 경우, 3 사분위 수가 30 에 있고, 최대 55 까지 늘어난다. 요약 데이터의 경우 3 사분위 수가 8 정도에 위치해 있다. 따라서 최대 길이 991 까지 고려했을 때 텍스트 제한길이 하이퍼파라미터를 각각 100, 8 로

지정했다. 다음으로 모든 문장에서 쓰인 단어의 개수를 측정하여 단어 집합의 크기를 6500 으로 정했다.

3) 모델 요약



Layer (type)	Output Shape	Param #	Connected to
input_9 (InputLayer)	[(None, 100)]	0	
embedding_3 (Embedding)	(None, 100, 128)	832000	input_9[0][0]
lstm_5 (LSTM)	[(None, 100, 256), (	394240	embedding_3[0][0]
input_10 (InputLayer)	[(None, None)]	0	
lstm_6 (LSTM)	[(None, 100, 256), (	525312	lstm_5[0][0]
embedding_4 (Embedding)	(None, None, 128)	832000	input_10[0][0]
lstm_7 (LSTM)	[(None, 100, 256), (	525312	lstm_6[0][0]
lstm_8 (LSTM)	[(None, None, 256),	394240	embedding_4[0][0] lstm_7[0][1] lstm_7[0][2]
attention_layer (AttentionLayer	((None, None, 256),	131328	lstm_7[0][0] lstm_8[0][0]
concat_layer (Concatenate)	(None, None, 512)	0	lstm_8[0][0] attention_layer[0][0]
dense_1 (Dense)	(None, None, 6500)	3334500	concat_layer[0][0]

Total params: 6,968,932
Trainable params: 6,968,932
Non-trainable params: 0

Figure 6

Figure 6 는 요약 모델 학습에 사용된 모델 summary 이다. 인코더는 LSTM 층 3 개를 쌓았고, 출력층은 Attention layer 를 결합해 어텐션 함수의 결과를 디코더의 hidden state 에 연결했다. 결과적으로 디코더의 시점 t 에서 단어를 예측하기 위해, 인코더의 모든 은닉 상태가 디코더의 현 시점의 은닉 상태와 각각 얼마나 유사한지를 반영할 수 있도록 하였다.

5. Evaluation

- Summarization

  1) Clearly analyze your result

  긍정문의 분류 결과 1376 개 중 일부는 다음과 같다.

song actually good tell im wrong
even half good new album
attention 90 ignore dont name yung static make music id really appreciate took two minute day listen friend good feeling ima blow up follow ill follow u back
look know everyone hate self promos im litterly 10 year old way put music could tell honest oppinion music would great thanks great day
check please im working real hard dis lmk think thank you
wonderful
heyyyy ya ll looking hard beat ? check producer come reposts

분류한 긍정문의 요약 내용은 다음과 같다 .

great album. great song. 50 cent. five stars. a must have. the best of the best. a good album. great. the best of the worst. not the best of the world. a classic. i love this song. great

부정문의 분류 결과 486 개 중 일부는 다음과 같다.

fuxxxx simp
ranciscoangel132795841 fxxx owner damn
scdrizzles wtf
shit period pooh
shit hard af

분류한 부정문의 요약 내용은 다음 과 같다.

one star. one. disappointed.

2) specify some novel or unique findings, patterns, insights you revealed

외국인 이용자들이남긴 SoundCloud 댓글 데이터에 너무 축약해서 쓴 말들이나 속어들이 많아 분류 한 결과물에서 알아 볼 수 없는 댓글의 형태가 다수 존재 하였다.

3) Evaluate and validate the model

rouge 는 자연어 생성 모델의 성능을 평가하기 위한 지표로, 모델이 생성한 요약본을 미리 만들어 놓은 참조본과 대조해 성능 점수를 계산한다.

```
summary.shpae= (8831, 1)
---Result of ROUGE-1---
recall =  0.20909916905274162
precision =  0.22836220888536615
rouge =  0.21300519081064884
```

Rouge-1 은 원래 요약문과 예측 요약문이 겹치는 unigram 수를 보는 지표로, 우리가 생성한 모델의 rouge1_recall, rouge1_precision, rouge1 의 결과를 도출하였다. Rouge-1 의 값은 약 0.21 의 결과가 나타났다.

```
summary.shpae= (8831, 1)
---Result of ROUGE-2---
recall =  0.14935643377495944
precision =  0.1538425244404182
rouge =  0.15032074959602987
```

rouge-2 는 원래 요약문과 예측 요약문이 겹치는 biagram 수를 보는 지표로, 우리가 생성한 모델의 rouge2_recall 과 rouge2_precision, rouge2 결과를 도출하였다. rouge-2 의 값은 약 0.15 의 결과가 나타났다.

4) Do the results of your model meet the goals you established?
기존 요약 모델 연구에서 rouge-1 로 모델을 평가하면 약 20-44%이상의 값을 기대하고, rouge-2 로 모델을 평가하면 약 10-21% 이상의 값을 기대한다. 우리가 생성한 모델의 rouge-1 의 값이 약 21%가 나오고, rouge-2 의 값이 약 15% 정도의 성능이 나왔기 때문에, 우리가 이루고자 하는 목표를 이루었다고 할 수 있다.

6.Reference

[1]pdf, "Neural Machine Translation By Jointly Learning To Align And Translate", 2016, Dzmitry Bahdanau, HyungHyun Cho Yoshua Bengio, arXiv:1409.0473v7 [cs.CL]

[2]pdf, "Sequence to Sequence Learning with Neural Networks", 2014, Ilya Sutskever, Oriol Vinyals, Quoc V.Le, arXiv:1409.3215