Sentiment Analysis Review

Peer Review

Reviewer: 김지원

Reviewee: 김산

증거사진 및 Comments

- 회고록이 존재함

회고하기

1. 과적합을 방지할 만한 여러가지 기법을 사용해보면 좋을 것 같다.(어떤 방식으로 해야하는 지 모름.)
2. earlystopping을 통해 여러번의 에포크를 실험하기 보다 한 번에 확인하는 것이 필요할 것 같다.
3. train / val / test의 데이터의 분포를 잘 파악하여 나누는 방법에 대해 생각해 볼 필요가 있는 것 같다.

- Subplots() 배운걸 적용해 그래프 시각화

성능 개선 그래프 시각화

[44]:
```
# 서브플롯을 사용하여 그래프 그리기
import matplotlib.pyplot as plt

# 훈련 손실 그래프
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(history1.history['loss'], 'r', label='model1 training loss')
plt.plot(history2.history['loss'], 'g', label='model2 training loss')
plt.title('Training Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.ylim(0.0, 1.0)

# 검증 손실 그래프
plt.subplot(1, 2, 2)
plt.plot(history1.history['val_loss'], 'r', label='model1 validation loss')
plt.plot(history2.history['val_loss'], 'g', label='model2 validation loss')
plt.title('Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.ylim(0.0, 1.0)

plt.tight_layout()
plt.show()

# 훈련 정확도 그래프
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(history1.history['accuracy'], 'r', label='model1 training accuracy')
plt.plot(history2.history['accuracy'], 'g', label='model2 training accuracy')
plt.title('Training Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
```

증거사진 및 Comments

- 3가지 이상의 모델을 적용하였으며, 이를 깔끔하게 보여주었다.

```
15]: #모델 평가

results = lstm.evaluate(X_test,y_test, verbose = 2)
print('lstm: ', results)

results = conv.evaluate(X_test,y_test, verbose = 2)
print('conv: ', results)

results = maxpool.evaluate(X_test,y_test, verbose = 2)
print('maxpool: ', results)
```

```
1527/1527 - 3s - loss: 0.6950 - accuracy: 0.8248
lstm:  [0.695035994052887, 0.824756383895874]
1527/1527 - 3s - loss: 1.3729 - accuracy: 0.7662
conv:  [1.3729016780853271, 0.7662327289581299]
1527/1527 - 2s - loss: 1.1709 - accuracy: 0.8140
maxpool:  [1.1708875894546509, 0.814030110836029]
```

- Step-by-step 으로 확인을 하였다

dataloader 함수에서 각 전처리를 하기위해 확인해야 할 것들.

1. 중복제거, nan값 제거

```
In [ ]: ## 복사해서 사용
train_data_c = train_data.copy()
test_data_c = test_data.copy()

# document 열과 label 열의 중복을 제외한 값의 개수
train_data['document'].nunique(), train_data['label'].nunique()
```

```
In [ ]: #nan값 확인
print(train_data_c.isnull().values.any())
```

```
In [ ]: #어떤 열에 있는지 개수 확인
print(train_data_c.isnull().sum())
```

```
In [ ]: #nan 값을 가진 샘플 출력
train_data_c.loc[train_data.document.isnull()]
```

```
In [ ]: #null값을 제거하고 잘 제거되었는지 확인

train_data_c = train_data_c.dropna(how = 'any') # Null 값이 존재하는 행 제거
print(train_data_c.isnull().values.any()) # Null 값이 존재하는지 확인
```

```
In [ ]: # 정규표현식을 활용하여 한글과 공백을 제외하고 모두 제거 + 살펴보기

train_data_c['document'] = train_data_c['document'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]", "", regex
train_data_c[:5]
```

```
In [ ]: # + 네이버 영화 리뷰는 영어 숫자 특수문자도 있어 이를 제거해줘야한다고 한다.
```