NLP Assignment 3

Data Processing

1. 구두점 제거
   a. Noise로 간주되는 구두점을 제거하였습니다.
   b. 이는, text의 실제 의미와 관계가 없는 요소로 판단하였으며, 토큰화의 일관성을 향상시켰습니다.

2. 소문자 변환
   a. 대소문자의 구분으로 똑같은 단어이지만, 대문자와 소문자 때문에 다르게 해석될 위험이 있어 소문자로 통일된 형식으로 변환하여 일관성을 부여했습니다.
   b. 이는, Dataset의 어휘 크기를 줄여줄 수 있습니다.

3. 토큰화
   a. text를 작은 단위로 분할하여 구조화할 수 있게끔 했습니다.
   b. Likelihood를 계산하기 위해 토큰화를 진행을 하기도 했습니다.

4. 불용어 제거
   a. 문법적으로는 중요한 단어이지만, text analysis에서는 큰 영향을 미치지 않는다고 생각했습니다.

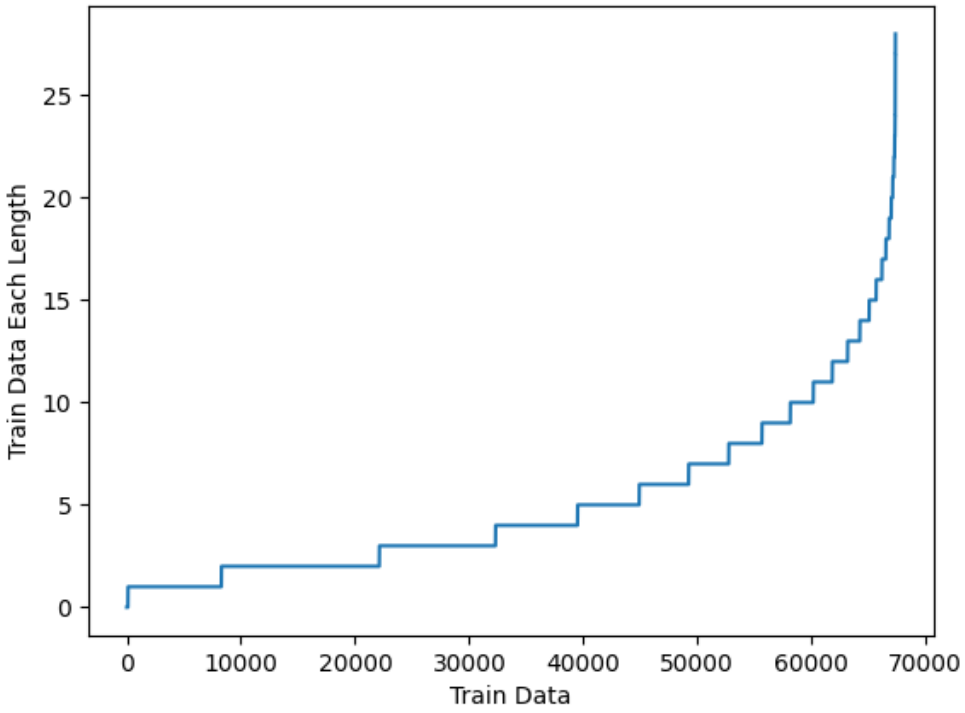
   b. 이는, 핵심 단어에 집중할 수 있도록 했습니다.

5. 표제어 추출
   a. 단어의 다양한 형태를 하나의 표제어로 통일하게 되면, 어휘의 일관성이 향상 된다고 합니다. 이를 통해, text를 더 효율적으로 분석할 수 있게 되었습니다.

   b. 어휘의 다양성을 줄였습니다. 이는 data의 dimension을 낮추기 때문에 inference과정에서 더 효율적입니다.

6. 단어의 길이

   a. network에 input datat를 넣기 위해서는, data의 크기(문장의 길이)는 동일해야 한다. train data의 최대 단어 길이는 28이였으며, 단어의 길이를 28을 기준으로 모든 vector를 28로 길이를 맞춰 주었다. (padding을 적용하였다.)



b. padding은 post로 정해주었습니다.

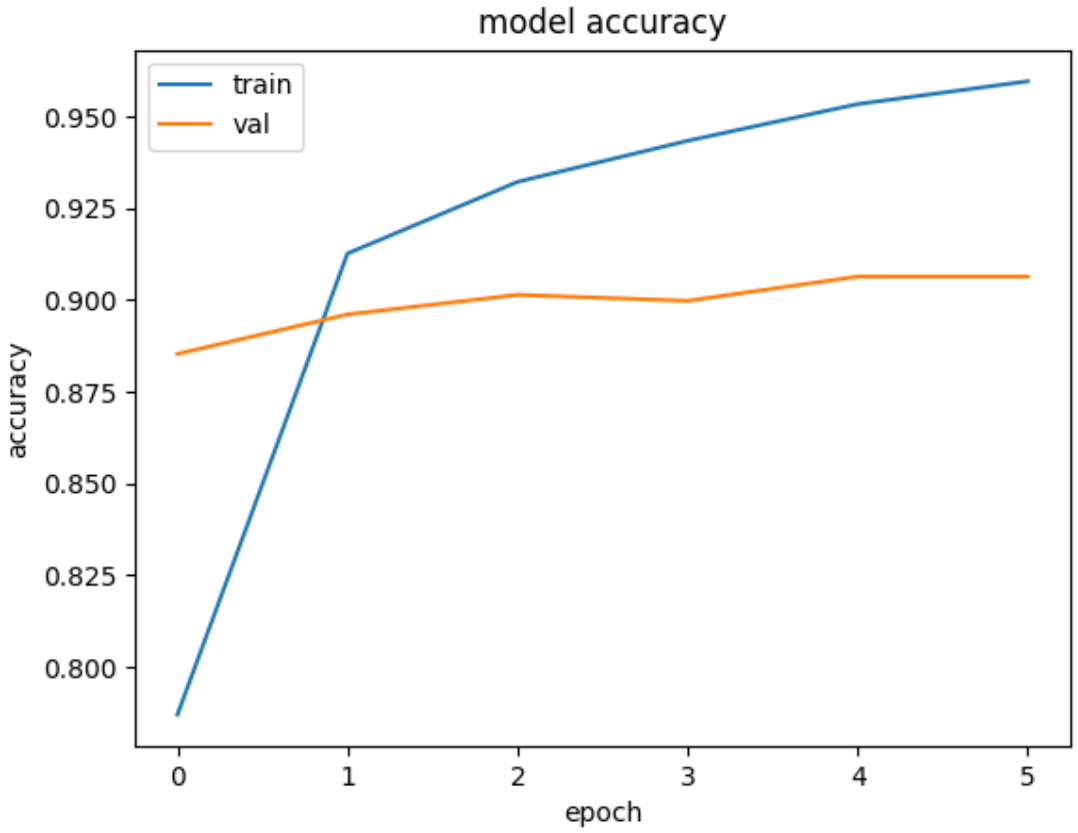
Modeling Layers

1. 모델은 LSTM을 썼으며, 20,000개의 word를 300dimension의 vector로 embedding을 했습니다.
2. LSTM의 output 크기는 128로 설정했습니다.
3. Dense layer를 쌓아, 2개의 output class이므로 2개의 output layer를 구축하여 softmax activation fucntion을 사용하였습니다.
4. optimizer는 Adam을 사용하였습니다.
5. one-hot-encoding을 사용하지 않았기 때문에, data의 label이 int형이여서, loss fucntion으로 sparse_categorical_crossentropy를 사용하였습니다.

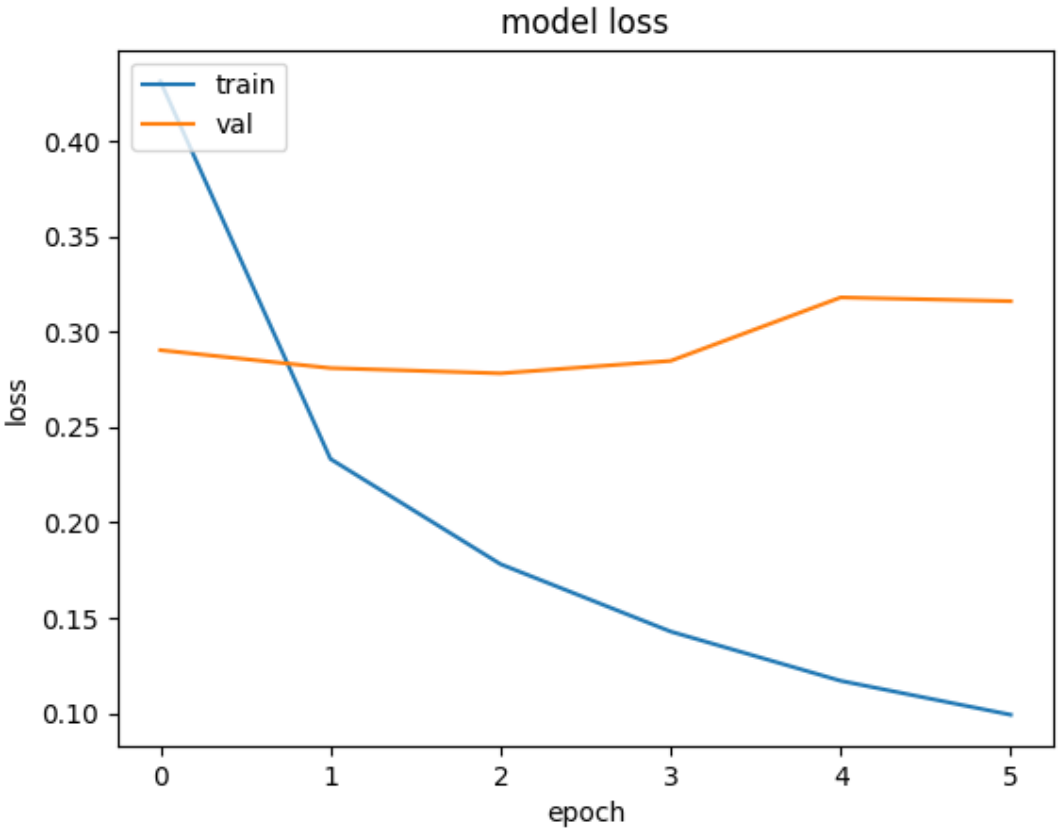
Metrics

dev.csv와 train.csv의 validation data로 평가지표를 나누어서 설명하겠습니다.

Analyze Validation Data of Train Data



위의 그래프를 보게 되면, train에서 validation data를 0.2로 나누어 validation accuracy를 평가한 지표인데, epoch가 증가할수록 accuracy가 증가하는 것을 볼 수 있다.



위의 그래프는, epoch에 따라 Loss의 값을 나타낸 그래프이다. 이 그래프도 accuracy 그래프와 마찬가지로 loss가 감소하는 것을 볼 수 있다.

Analyze Test data of Analysis

	precision	recall	f1-score	support
0	0.81	0.77	0.79	428
1	0.79	0.83	0.81	444
accuracy			0.80	872
macro avg	0.80	0.80	0.80	872
weighted avg	0.80	0.80	0.80	872

이는, classification_report라는 함수를 통해서 0(Negative)와 1(Positive)일 때의 각각의 precision, recall, f1-score의 값을 볼 수 있다.

각 class마다 precision과 recall이 balance하게 output 값이 나왔기 때문에 f1-score도 balance한 output이 나오게 되었다.

따라서, model이 안정적으로 balance하게 predict하고 있다는 것을 볼 수 있다.