LSTM을 이용한 Text 속 감정 분류

Park Ju Ho:An undergraduate in statistics
Jung Sang Hoon:Mentor

1. INTRODUCTION

- 이 연구는 RNN의 한 종류인 LSTM을 이용하여 영어 자연어를 처리하는 기본적인 방법을 보여주고 있다.
- 과정은 크게 전처리와 모델 학습으로 나뉘고 최종적으로 loss와 accuracy를 확인한다.
- 전처리는 결측치처리,데이터 분리,토큰화,인덱싱,패딩의 순서로 이루어진다.
- 모델은 한 층의 RNN으로 가장 간단한 LSTM모델을 구현하였다.

2. METHODS

- Pre-processing

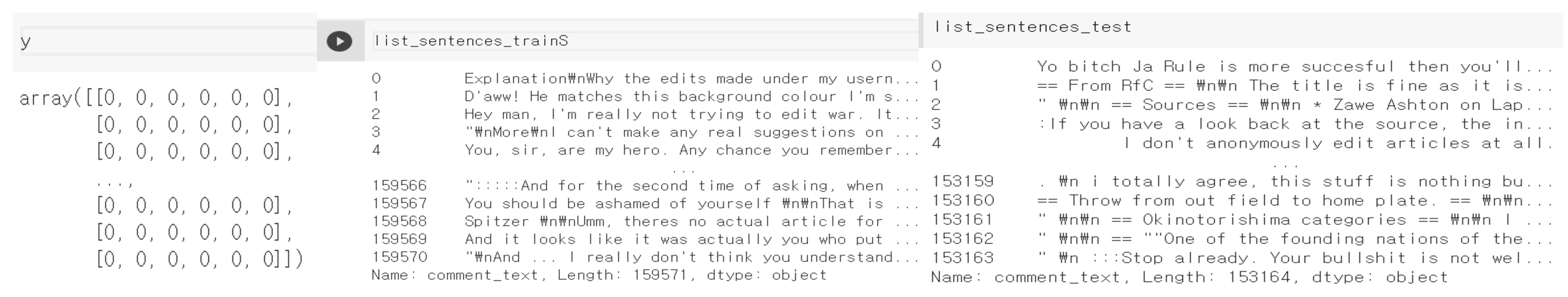
1. Missing Value` Processing

```
train.isnull().any()

id              False
comment_text    False
toxic           False
severe_toxic    False
obscene         False
threat          False
insult          False
identity_hate   False
dtype: bool
```

```
test.isnull().any()

id              False
comment_text    False
dtype: bool
```

Train Data와 Test Data 모두 결측치가 없으므로 결측치를 처리 할 필요는 없음

2. Data Isolation



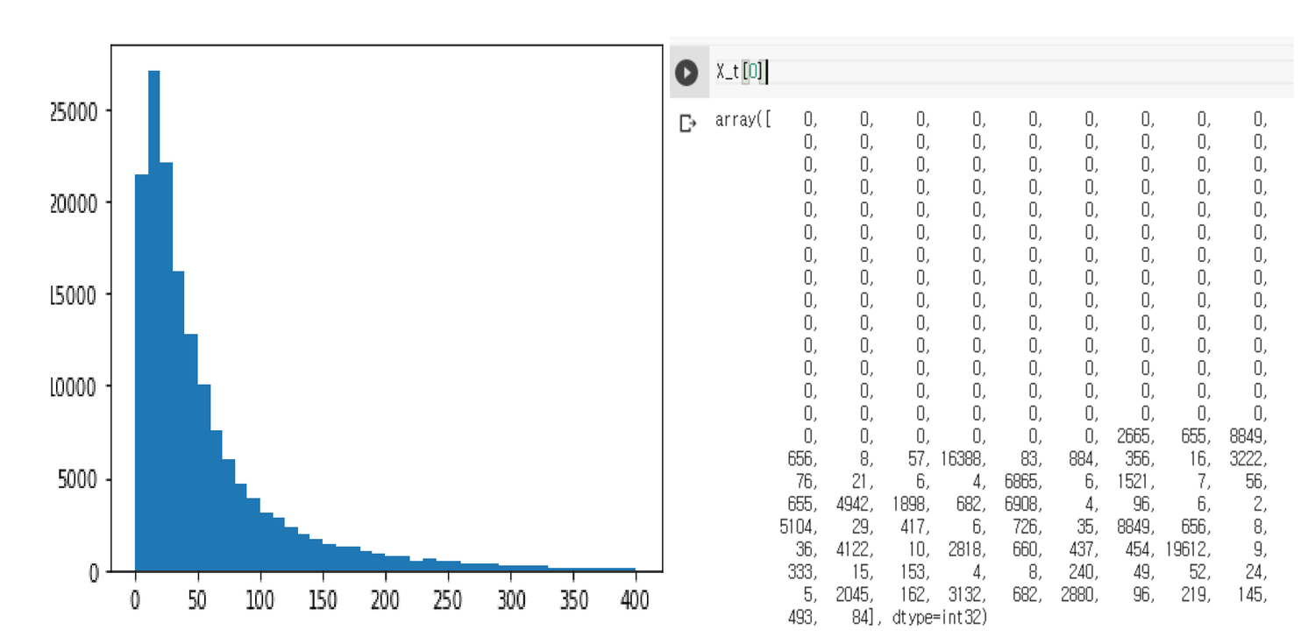
Y=감정 유무의 이진 데이터, train&test = text데이터

3. Tokenization & Indexing

```
print(list_tokenized_train[:1])

[[688, 75, 1, 126, 130, 177, 29, 672, 4511, 12052, 1116,
```

문장을 토큰화 시킨 후 각 단어를 인덱싱 시켜 문장을 인덱싱 번호로 바꾼 작업

4. Padding

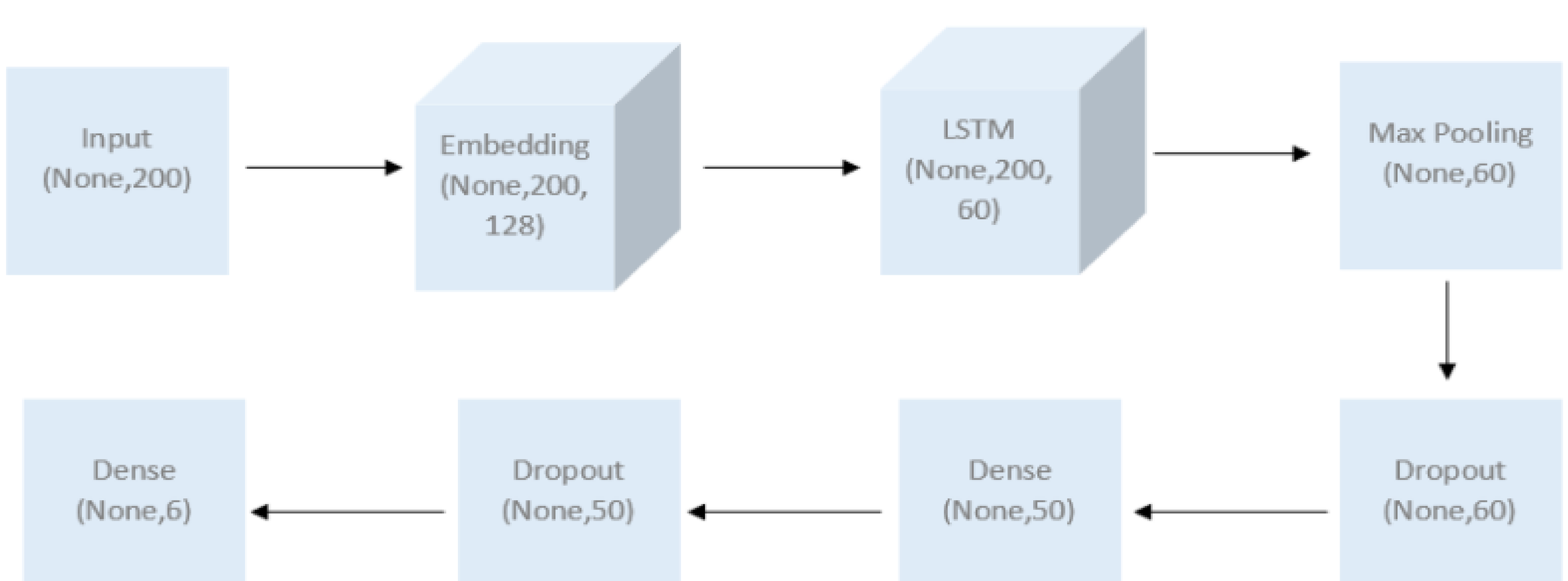


1. Maxlen을 40~50(그래프상 평균)으로 잡는다 => 정보가 손실됨(효율성 우선)
2. Maxlen을 더 크게 잡는다 => 학습이 오래 걸림(정확도 우선)

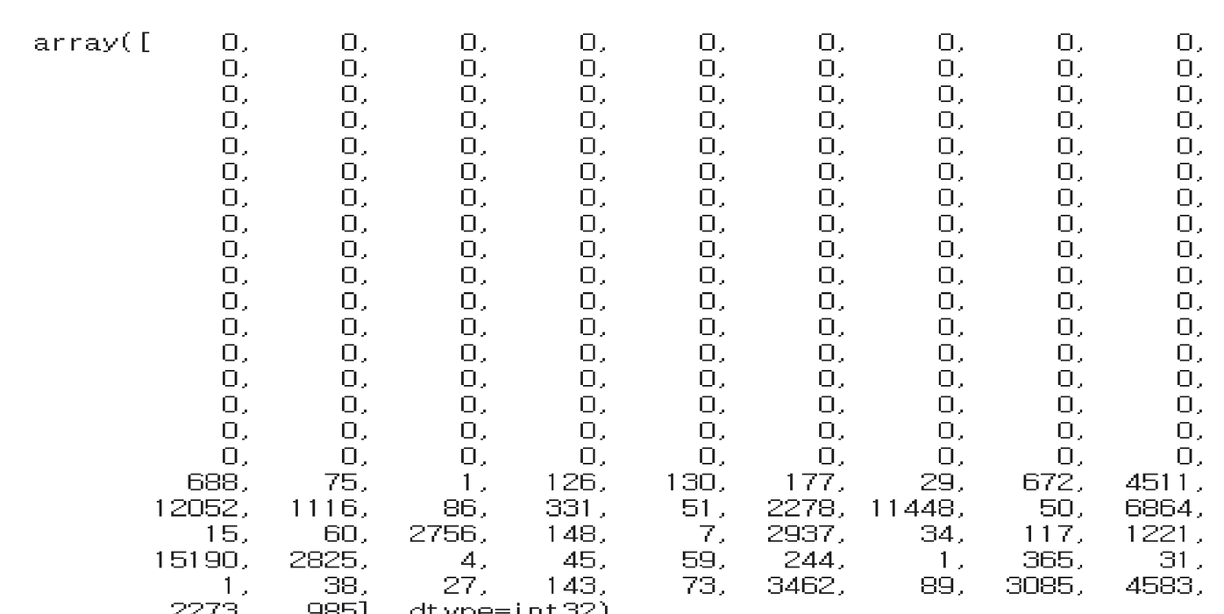
여기선 정확도를 좀 더 높이기 위하여 2번을 선택하여 maxlen을 200으로 잡고 padding을 진행시킴

- Modeling

0. 모델링 설계도

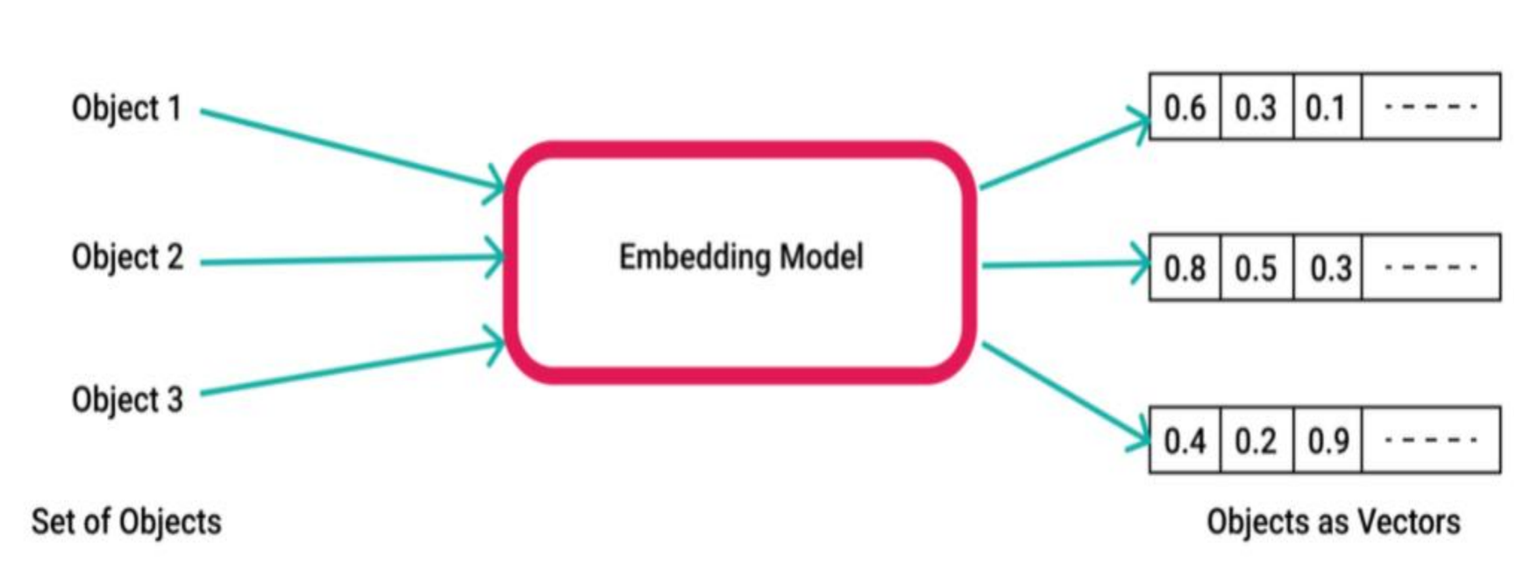


1. Input



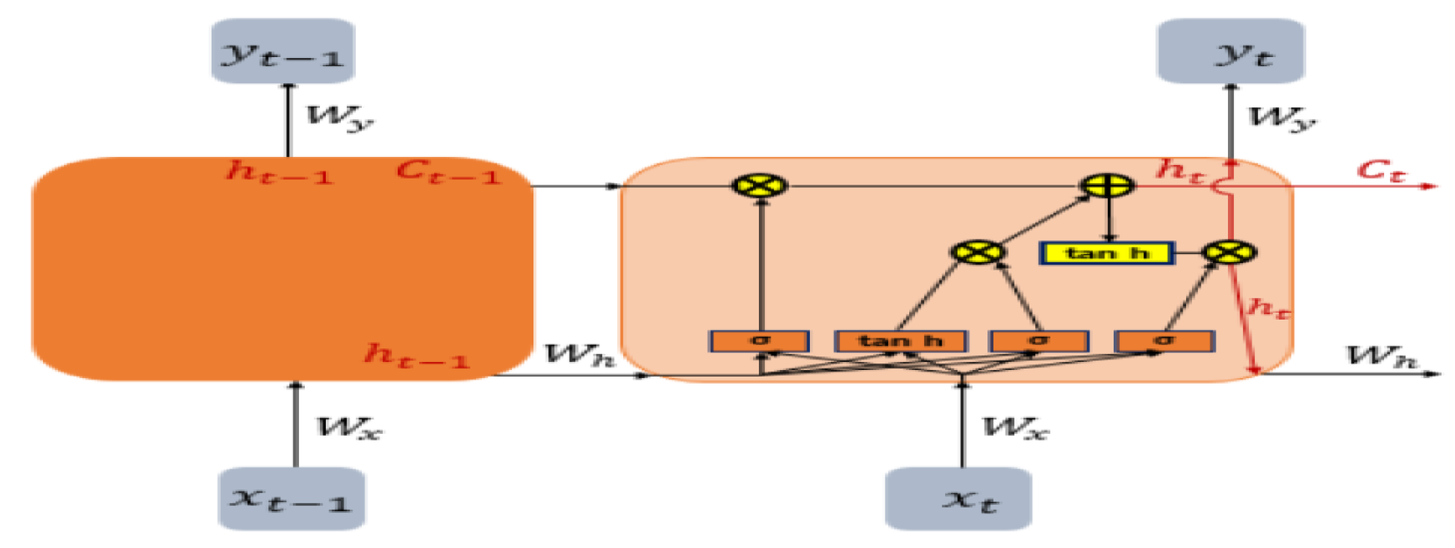
Maxlen의 길이가 200이므로 input의 차원은 200으로 설정함

2. Embedding



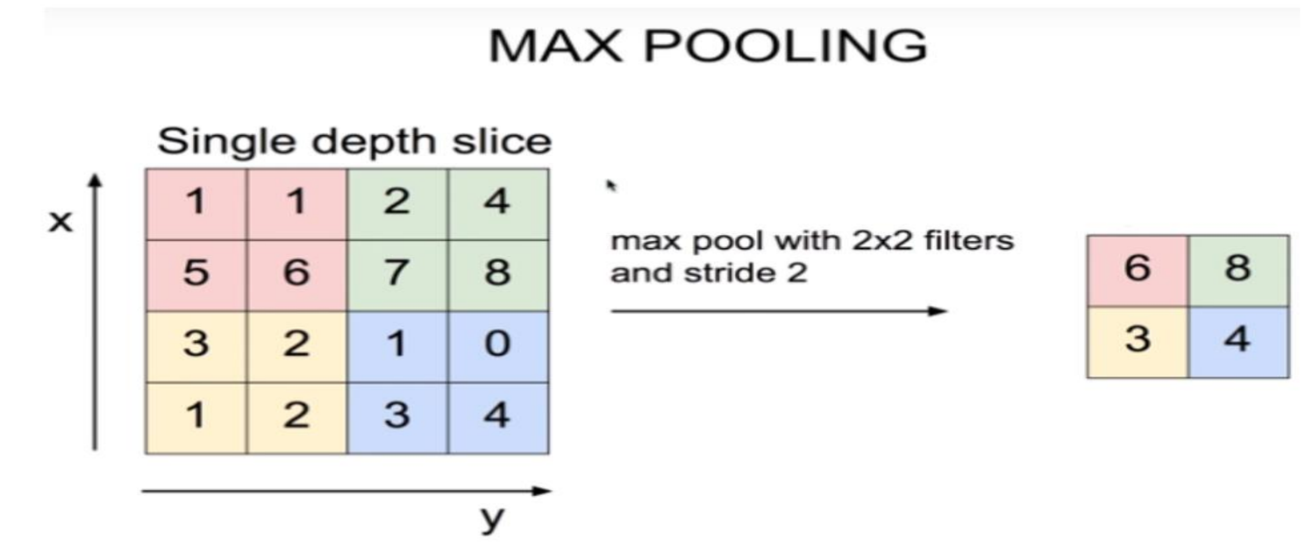
One-hot 인코딩 된 벡터를 임베딩함으로써 One-hot의 단점인 높은 차원을 해소할 수 있음
뿐만 아니라 임베딩 된 벡터를 이용 벡터의 성질(거리 등)을 그 대로 적용 시킬 수 있는 장점이 있음

3. LSTM



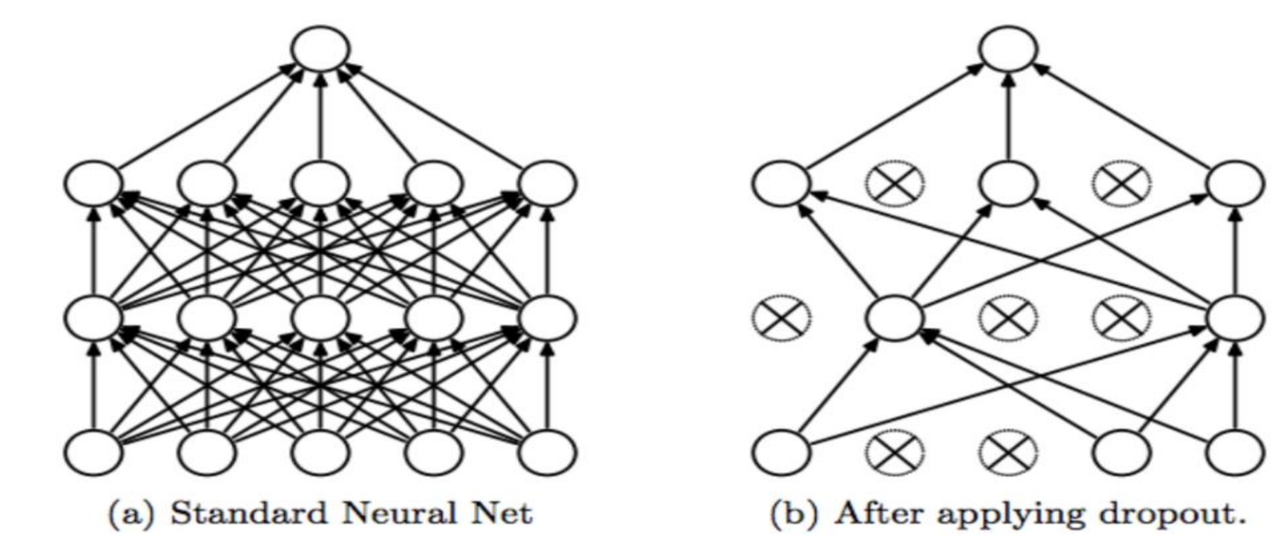
Lstm이란? 기존 바닐라 RNN의 단점인 장기 의존성 문제를 해결한 모델
은닉층에 3개의 게이트를 추가하여 불필요한 기억은 지우고 기억해야 할 것들만 가중치를 줌

4. Max Pooling



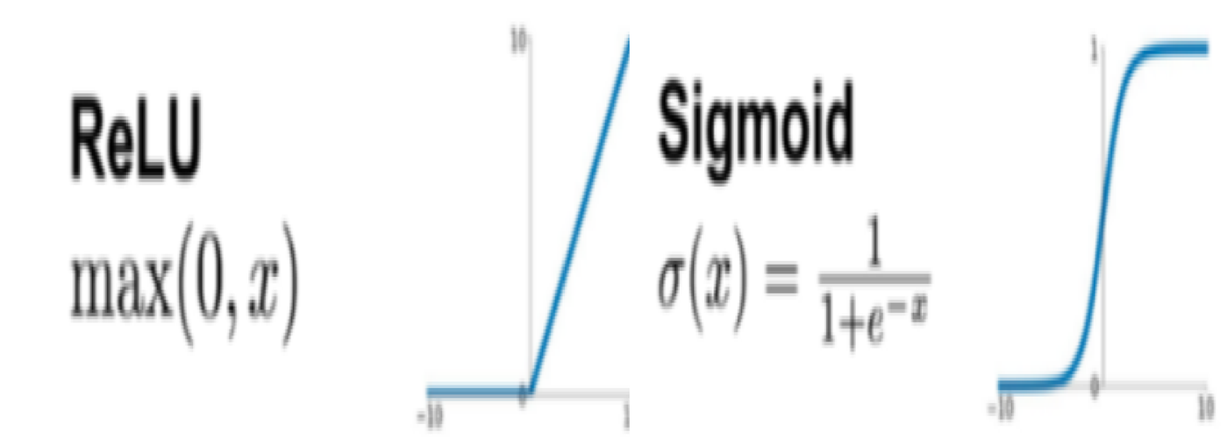
임베딩으로 인해 높아진 차원을 Pooling을 통하여 감소시켜 줌

5. Dropout



Dropout 즉 weight값 중 일부를 부분적으로 생략하여 과적합을 예방해준다.

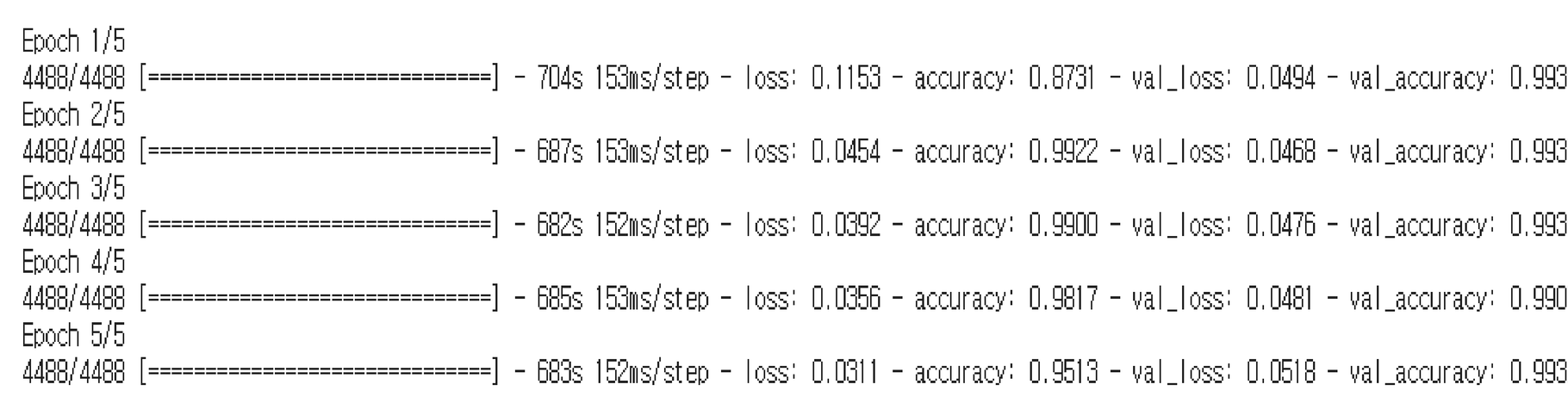
6. Dense1&2



Dense1:은닉층에 주로 사용되는 ReLu 함수를 적용
Dense2:이진분류 문제이므로 Sigmoid를 활용하고 출력 해야하는 차원에 맞춰 6으로 설정

3. CONCLUSION

- Result

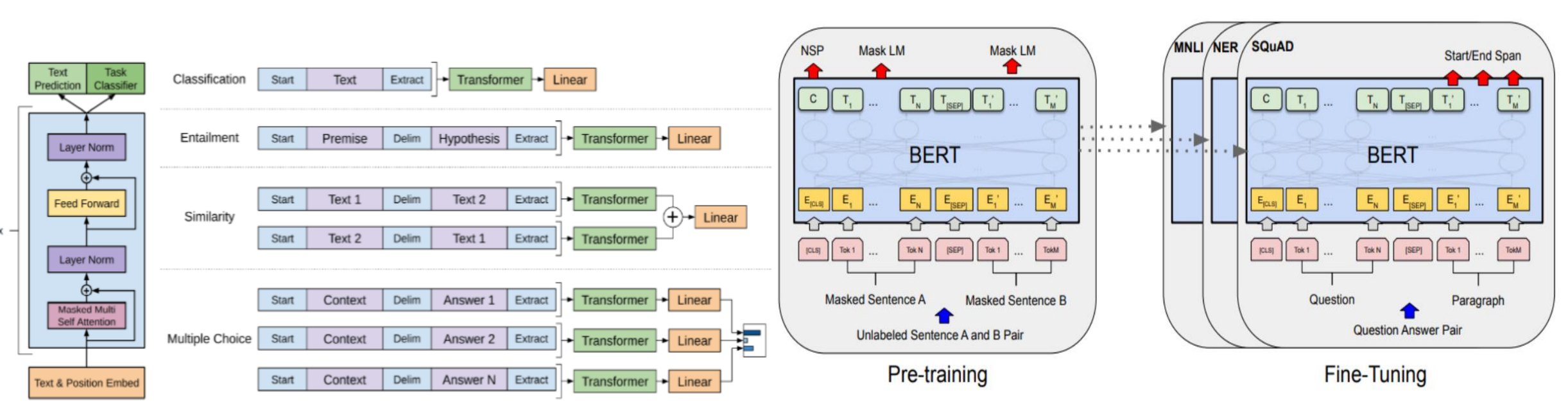


```
4987/4987 [==============================] - 144s 29ms/step - loss: 0.0283 - accuracy: 0.9938
[0.028314122930169106, 0.9937582612037659]
```

최종 손실값:0.0283, 정확도:0.9938로 학습이 매우 잘 된 것을 확인 할 수 있다.
하지만 너무 높은 정확도로 보아 과적합에 우려가 있다.

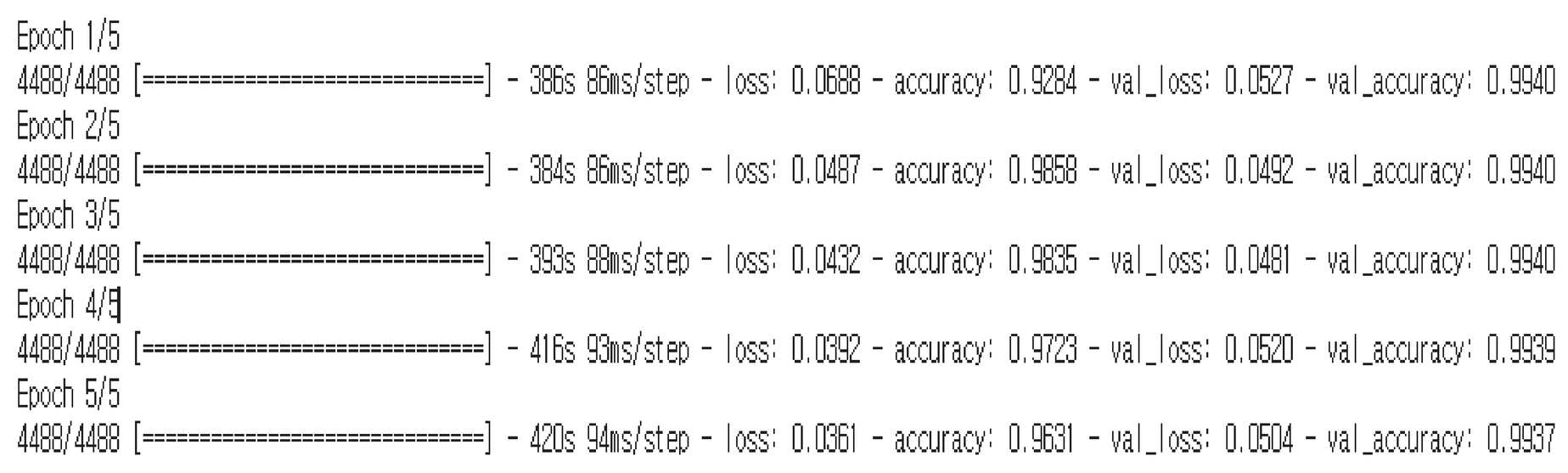
- Improving

1. Pre-trained model



Gpt-1(좌),Bert(우)처럼 엄청난 데이터를 바탕으로 훈련된 모델을 이용하여 성능을 향상시킬 수 있다.

2. .Change Model



```
4987/4987 [==============================] - 82s 16ms/step - loss: 0.0311 - accuracy: 0.9940
[0.031080698594445095, 0.9940152168273926]
```

모델을 LSTM이 아닌 SimpleRNN으로 바꾸어 모델링을 진행하였다.
최종 손실값:0.031, 정확도 0.9940으로 LSTM과 큰 차이를 보여주고 있지 않다.

3.Change Parameter

```
4987/4987 [==============================] - 331s 66ms/step - loss: 0.0300 - accuracy: 0.9712
[0.030006036162376404, 0.9711915254592896]
```

LSTM출력층의 레이어를 128로 높이고 과적합이 일어나는 것 같아 Dropout의 비율을 0.2로 상승시켰다.
그리고 epoch을 20으로 높여 보았다.
최종 손실값:0.03,정확도0.9712로 정확도가 살짝 떨어진 것을 보아 과적합을 적당히 예방하였다.
손실 역시 낮은 수치를 보여주고 있다.

4.Callback

Callback 함수를 활용하여 val_loss의 향상이 2번 없을 경우 학습을 중지하는 것
모델 학습의 효율성의 좋아짐

- Conclusion

1. 이 연구를 통하여 가장 간단한 형태의 자연어 분류 모델을 만들어 봄으로써 모델의 자연어 처리 과정과 및 향상 방안에 대하여 고민 할 수 있음
2. 고민한 향상 방안은 크게 4가지이며 가장 좋은 방법은 Pre-trained model로 예상되나 장비의 문제로 실행하지 못 함
3. 그것을 제외하곤 변수 변환 및 모델 변환을 적용해 보았으나 큰 성과를 거두진 못하였고 callback은 통한 효율성은 확실히 획득함