



# [2주차] 기본과제: 주어진 문장에서 나올 다음 단어를 예측하는 모델 구현



이번 과제에서는 Transformer를 last word prediction이라는 task에 적용합니다. Last word prediction은 Token list가 주어졌을 때, 다음으로 오는 token을 예측하는 task로, 추후 등장할 LLM의 핵심입니다.

## 준비

Transformer 실습을 진행한 notebook 위에서 진행해주시면 됩니다:

[drive.google.com](https://drive.google.com)

[https://drive.google.com/file/d/1vh4ALpRVicq9hdonHYiLTIsPhY6LLZAE/view?usp=share\\_link](https://drive.google.com/file/d/1vh4ALpRVicq9hdonHYiLTIsPhY6LLZAE/view?usp=share_link)

## 목표

## □ Last word prediction dataset 준비

- 기존의 IMDB dataset을 그대로 활용하고, `collate_fn` 을 다음과 같이 수정하면 됩니다:

```
from torch.nn.utils.rnn import pad_sequence
def collate_fn(batch):
    max_len = 400
    texts, labels = [], []
    for row in batch:
        labels.append(tokenizer(row['text'], truncation=True, max_length=max_len).input_ids[-2])
        texts.append(torch.LongTensor(tokenizer(row['text'], truncation=True, max_length=max_len).input_ids[:-2]))
    texts = pad_sequence(texts, batch_first=True, padding_value=tokenizer.pad_token_id)
    labels = torch.LongTensor(labels)
    return texts, labels
```

## □ Loss function 및 classifier output 변경

- 마지막 token id를 예측하는 것이기 때문에 binary classification이 아닌 일반적인 classification 문제로 바뀝니다. MNIST 과제에서 했던 것 처럼 loss와 `TextClassifier` 의 출력 차원을 잘 조정하여 task를 풀 수 있도록 수정하시면 됩니다.

## □ 학습 결과 report

- 기존 Transformer 실습에서 사용한 모델로 last word prediction을 학습하고 학습 경과를 report하면 됩니다.

## 제출자료

제약 조건은 전혀 없으며, 위의 사항들을 구현하고 epoch마다의 train accuracy와 test accuracy가 print된 notebook을 public github repository에 업로드하여 공유해주시면 됩니다(반드시 출력 결과가 남아있어야 합니다!!).

## 과제 제출 방법 상세

- 과제 목표에 대한 체크 리스트에 따라 Colab 노트북에서 과제를 수행합니다.

## 2. 노트북 중간 중간 텍스트 블록(마크 다운)을 추가해 아래 항목을 표기합니다.

a. 텍스트는 모두 ##(제목2) 형식으로 기재합니다.



b. 아래 3가지를 형식에 맞추어 기재합니다.

항목	제목 형식	내용	예시
수행한 부분	[MY CODE]	수행한 내용에 대한 설명	[MY CODE] Test data 준비하기
출력 결과가 남은 부분	[LOG]	출력 로그 설명	[LOG] 학습 과정에서의 Epoch별 손실 값 출력
피드백 요청 부분	[FEEDBACK]	질문 또는 개선 요청 내용을 간결히 정리	[FEEDBACK] 정확도를 더 높이려면 어떻게 해야 할지 궁금합니다!

## 3. 개인 GitHub에 Public Repository를 하나 생성합니다.

a. 반드시 접근이 가능하도록 Public으로 생성해 주세요!

## 4. 과제 진행이 모두 완료되면 GitHub에 사본을 가져옵니다.

a. 파일 > GitHub에 사본 저장 > 저장소 선택 > 확인

5. GitHub으로 가져온 jovnph 파일 링크를 과제 제출 페이지에 제출합니다.