

# News Topic Classification

: 월간 데이콘 뉴스 토픽 분류 AI 경진대회

2023 Summer Vacation DL Class

Team 1

강동현 (산업경영공학부 19)  
김나연 (통계학과 19)

# Contents

01. 프로젝트 소개

02. 데이터 EDA

03. 자연어 전처리

04. 모델링

05. 결과



# 01. 프로젝트 소개

The screenshot shows the DAICON website interface. At the top, there is a navigation bar with links to '커뮤니티', '대회', '학습', '랭킹', and '더보기'. On the right side of the header are icons for search, globe, notifications, and a red 'Home' button. Below the header, a banner for the competition is displayed, featuring a collage of newspaper clippings and the text '월간 데이콘 뉴스 토픽 분류 AI 경진대회'. The banner includes details such as '알고리즘 | NLP | 분류 | 자연어 | Accuracy', '상금 : 500,000 D-point', '기간 : 2021.06.30 ~ 2021.08.09 17:59', and '참여자 수 : 1,575명'. A blue button labeled '제출 XP 획득!' is visible. Below the banner, there is a section titled 'YNAT (주제 분류를 위한 연합 뉴스 헤드라인)' with a sub-section '데이터 세트를 활용해 주제 분류 알고리즘을 개발하는 프로젝트'. The main content area features a sidebar with sections like '개요', '규칙', '일정', '상금', and '동의사항'. The main content area also contains sections for '1. 배경' and '2. 대회 개요', along with various charts and tables related to the competition.

**YNAT (주제 분류를  
위한 연합 뉴스 헤드라인)  
데이터 세트를 활용해  
주제 분류 알고리즘을  
개발하는 프로젝트**

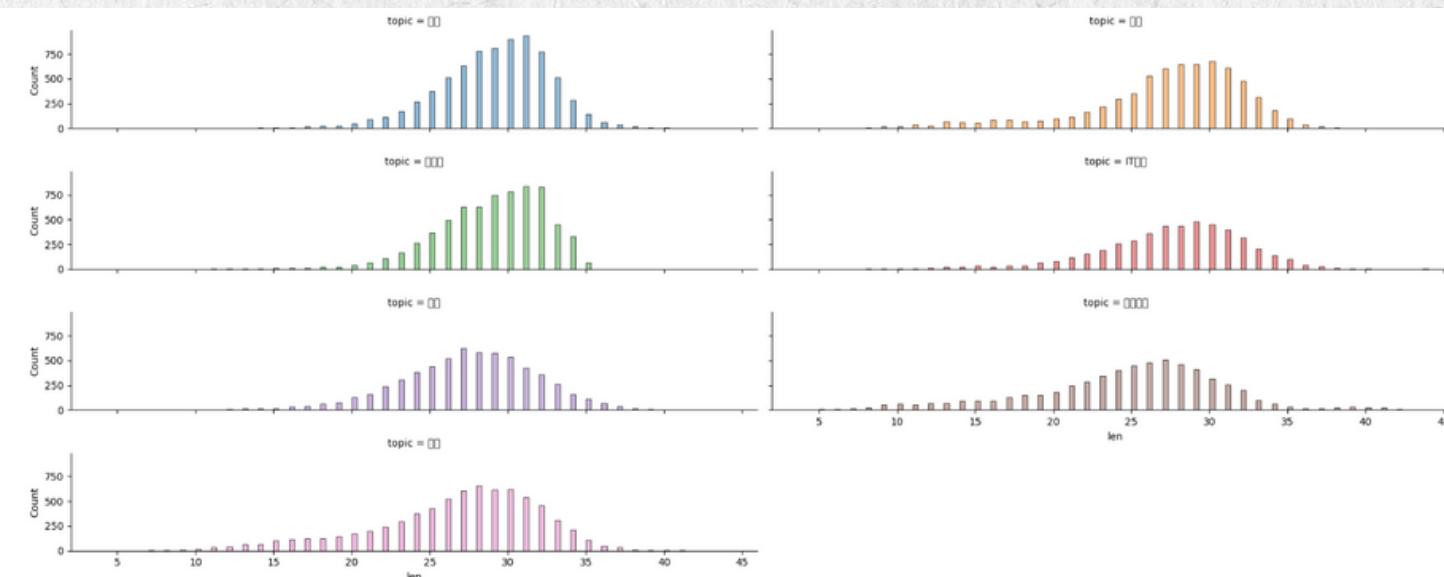
**2021년에 열렸으며,  
참고할 수 있는 코드나  
자료가 풍부한 DAICON의  
자연어처리 대회**

## 02. 데이터 EDA

### train dataset

index		title	topic_idx
0	0	인천→핀란드 항공기 결항…휴가철 여행객 분통	4
1	1	실리콘밸리 넘어서겠다…구글 15조원 들여 美전역 거점화	4
2	2	이란 외무 긴장완화 해결책은 미국이 경제전쟁 멈추는 것	4
3	3	NYT 클린턴 측근韓기업 특수관계 조명…공과 사 맞물려종합	4
4	4	시진핑 트럼프에 중미 무역협상 조속 타결 희망	4
5	5	팔레스타인 가자지구서 16세 소년 이스라엘군 총격에 사망	4
6	6	인도 48년 만에 파키스탄 공습…테러 캠프 폭격종합2보	4

### 문장별 길이 시각화



title : 기사 제목  
topic\_idx : 라벨 인덱스

결측치가 존재 X  
중복데이터 존재 X

세계 : 7629  
사회 : 7362  
스포츠 : 6933  
정치 : 6751  
경제 : 6222  
생활문화 : 5933  
IT과학 : 4824

### topic\_dict

topic	topic_idx
IT과학	0
경제	1
사회	2
생활문화	3
세계	4
스포츠	5
정치	6

## 03. 자연어 전처리

### 품사 태깅 및 어간 추출

- Komoran, OKT, KKMA 중 OKT 사용

### 데이터 정제

- 특수문자, 단위, 대~소문자 전환 등
- 기본 정제

### 불용어 처리

- 불용어들을 stop\_words에 정의 후, 제거

### 단어 분포 시각화

- wordcloud 패키지 활용, Topic별 단어 분포 시각화

### (Roberta) Tokenize

- keras 의 Tokenizer 패키지 및 DataLoader 적용

### (양방향 LSTM) 정수 인코딩

- 기본 전처리 완료 데이터 셋의 통계량을 통해 파라미터 설정, 인코딩 진행

### (양방향 LSTM) 패딩

- 문장의 최대길이 만큼 패딩

### (양방향 LSTM) 원 - 핫 인코딩

- 원핫 인코딩 실시 후, numpy 형태로 저장

# 03. 자연어 전처리

## a. 품사 태깅 및 어간 추출

```
kkma      : [ ('시진', 'NNG'), ('핑', 'MAG'), ('트럼프', 'NNG'), ('에', 'JKM'), ('중미', 'NNG'), ('무역', 'NNG') ]
okt       : [ ('시진핑', 'Noun'), ('트럼프', 'Noun'), ('에', 'Josa'), ('중미', 'Noun'), ('무역', 'Noun'), ('협상', 'Noun') ]
komoran   : [ ('시진핑', 'NNP'), ('트럼프', 'NNP'), ('에', 'JKB'), ('중', 'NNB'), ('미', 'NNP'), ('무역', 'NNG') ]
```

뉴스 기사 도메인의  
특성상, Noun만  
정확하게 추출 가능  
하며 속도 측면에서  
이점이 확실한 OKT 활용

## b. 데이터 정제 및 한자어 처리

## c. 불용어 처리

index	title	topic_idx	clean_title
0	인천→핀란드 항공기 결항…휴가철 여행객 분통	4	인천 핀란드 항공기 결항 휴가 여행객 분통
1	실리콘밸리 넘어서겠다…구글 15조원 들여 美전역 거점화	4	실리콘밸리 넘어서다 구글 조원 들이다 미국 전역 거점
2	이란 외무 긴장완화 해결책은 미국이 경제전쟁 멈추는 것	4	이란 외무 긴장 완화 해결 미국 경제 전쟁 멈추다
3	NYT 클린턴 측근韓기업 특수관계 조명…공과 사 맞물려종합	4	nyt 클린턴 측근 한국 기업 특수 관계 조명 공과 사 맞다 물리다 종합
4	시진핑 트럼프에 중미 무역협상 조속 타결 희망	4	시진핑 트럼프 중미 무역 협상 조속 타결 희망
5	팔레스타인 가자지구서 16세 소년 이스라엘군 총격에 사망	4	팔레스타인 가자지구 소년 이스라엘군 총격 사망
6	인도 48년 만에 파키스탄 공습…테러 캠프 폭격종합2보	4	인도 파키스탄 공습 테러 캠프 폭격 종합
7	美대선 TV토론 음담패설 만회실패 트럼프…사과 대신 빌클린턴 공격해 역효과	4	미국 대선 tv 토론 음담패설 만회 실패 트럼프 사과 대신 빌다 클린턴 공격 역효과
8	푸틴 한반도 상황 진전 위한 방안 김정은 위원장과 논의	4	푸틴 한반도 상황 진전 방안 김정은 위원장 논의
9	특검 면죄부 받은 트럼프 스캔들 보도 언론 맹공…국민의 적	4	특검 면죄부 받다 트럼프 스캔들 보도 언론 맹공 국민

```
## train_df 데이터셋 전처리
train_df["clean_title"] = train_df["title"].apply(lambda x: pos_cleaner(x))
train_df["clean_title"] = train_df["clean_title"].apply(lambda x: text_preprocessor(x))
train_df["clean_title"] = train_df["clean_title"].apply(lambda x: chinese_to_korean_replacer(x))
train_df["clean_title"] = train_df["clean_title"].apply(lambda x: del_stop_words(x))

train_df.head(10)
```

a, b, c 각각 함수로 정의 후 train dataset의  
"clean\_title" column을 새롭게 선언

## 03. 자연어 전처리

# WordCloud : topic별 단어 분포 시각화



## 03. 자연어 전처리

**추가 전처리의 필요성 : 단어 분포 파악 결과, 여러 토픽에서 공통적으로 등장하는 단어 존재**

0 : IT science



1 : Economics



2 : Society



3 : Culture



4 : World



5 : Sports



6 : Politics



## 03. 자연어 전처리

### 전체 데이터셋의 단어 중, 토픽마다 등장하는 단어들을 파악해주는 함수

```
def cnt_duplicating_topics(seq):
    result = []
    all_topics_top_20 = [tags_keys_0, tags_keys_1, tags_keys_2, tags_keys_3, tags_keys_4, tags_keys_5, tag
    for i in range(30):
        cnt = 0
        common_word = seq[i][0]
        for j in all_topics_top_20:
            if common_word in j:
                cnt += 1
            if cnt >= 3:
                result.append(common_word)
    return result
```

[ '종합', '한국' ]

위의 두 단어 이외에도  
토픽별 최다 빈도 단어 중 필요없는  
단어를 불용어 리스트에 추가 후 제거  
ex. "억원", "조원" ...

### 기본 전처리 후, 최종적으로 사용할 데이터셋

	index	title topic_idx	clean_title	title_list
0	0	인천→핀란드 항공기 결항…휴가철 여행객 분통	인천 핀란드 항공기 결항 휴가 여행객 분통	[인천, 핀란드, 항공기, 결항, 휴가, 여행객, 분통]
1	1	실리콘밸리 넘어서겠다…구글 15조원 들여 美전역 거점화	실리콘밸리 넘어서다 구글 들이다 미국 전역 거점	[실리콘밸리, 넘어서다, 구글, 들이다, 미국, 전역, 거점]
2	2	이란 외무 긴장완화 해결책은 미국이 경제전쟁 멈추는 것	이란 외무 긴장 완화 해결 미국 경제 전쟁 멈추다	[이란, 외무, 긴장, 완화, 해결, 미국, 경제, 전쟁, 멈추다]
3	3	NYT 클린턴 측근韓기업 특수관계 조명…공과 사 맞물려종합	nyt 클린턴 측근 기업 특수 관계 조명 공과 맞다 물리다	[nyt, 클린턴, 측근, 기업, 특수, 관계, 조명, 공과, 맞다, 물리다]

## 03. 자연어 전처리

### Roberta 모델을 위한 Tokenize

: transformers의 AutoTokenizer 패키지 활용

```
class TVDataset(Dataset):
    def __init__(self, csv_file, model_name):
        self.dataset = csv_file
        self.tokenizer = AutoTokenizer.from_pretrained(model_name)

    def __len__(self):
        return len(self.dataset)

    def __getitem__(self, idx):
        row = self.dataset.iloc[idx, 2:4].values
        text = str(row[1])
        y = row[0]

        inputs = self.tokenizer(
            text,
            return_tensors='pt',
            truncation=True,
            max_length=14,
            pad_to_max_length=True,
            add_special_tokens=True
        )

        input_ids = inputs['input_ids'][0]
        attention_mask = inputs['attention_mask'][0]
        y = torch.tensor(y) # 이걸 꼭해줘야한다...
        return input_ids, attention_mask, y
```

: 패딩 추가 등의 옵션을 두고, dataloader 생성을 위한 기본 조건 선언

### 양방향 LSTM 모델을 위한 Tokenize

: keras의 Tokenizer 및 to\_categorical 패키지 활용

```
before : [ 7 1 1 1 1 2 92 1 1 0 0 0 0 0 ]
before length : 14
after : [[0. 0. 0. ... 0. 0. 0.]
          [0. 1. 0. ... 0. 0. 0.]
          [0. 1. 0. ... 0. 0. 0.]
          ...
          [1. 0. 0. ... 0. 0. 0.]
          [1. 0. 0. ... 0. 0. 0.]
          [1. 0. 0. ... 0. 0. 0.]]
after length : 14
```

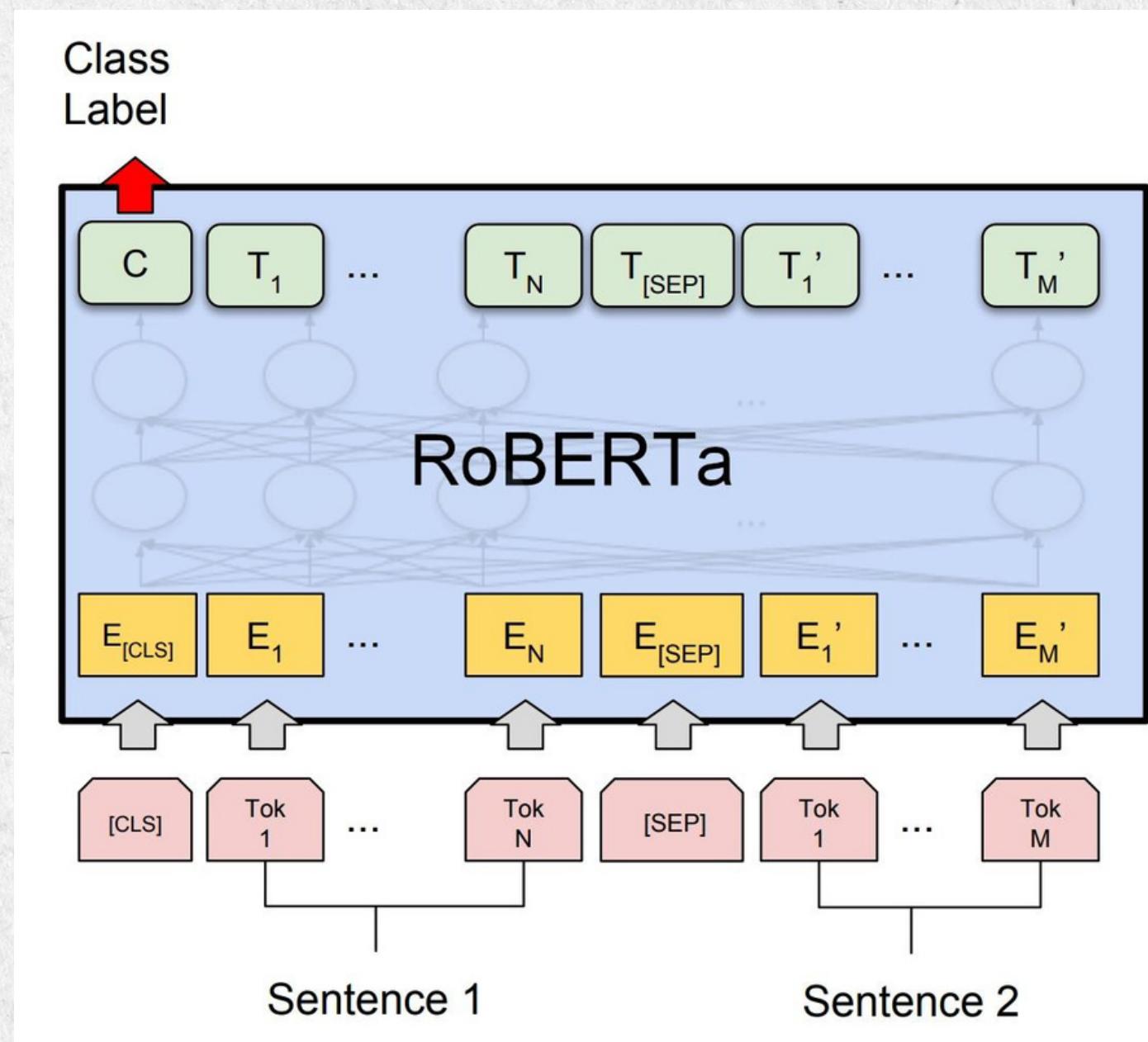
: 원 핫 인코딩 결과물

: 인코딩 완료 후, 모델 활용을 위해 numpy 파일화

```
np.save('/content/drive/MyDrive/23-2 Kubig Contest/KUBIG-DLcontest/dataset/X_train', train_inputs)
np.save('/content/drive/MyDrive/23-2 Kubig Contest/KUBIG-DLcontest/dataset/Y_train', train_labels)
np.save('/content/drive/MyDrive/23-2 Kubig Contest/KUBIG-DLcontest/dataset/X_test', test_inputs)
```

## 04. 모델링

### 모델 1 : RoBERTa (large / small / base)



: BERT 모델을 기반으로 하는 NLP 모델

- 기존의 BERT 모델이 undertrained 된 사실에 초점
- BERT 논문에서 hyper parameter에 대한 실험이 제대로 진행되지 않았기 때문에 기존의 model을 유지하며 학습 단계의 parameter를 조정하여 성능을 높이는 방안을 채택

: BERT 모델에 비해 추가되는 부분

- dynamic masking 기법 적용
- NSP 제거
- 더 긴 시퀀스를 이용해 모델 학습
- 더 많은 데이터, 더 큰 배치로 모델 학습

~ klue roberta model은 32000개의 vocab size를 보유하며, 높은 성능을 입증했으므로 해당 프로젝트에서 사용하기에 적합

## 04. 모델링

### 모델 1 : RoBERTa (large / small / base)

```

model_roberta_large = "klue/roberta-large"

train_1 = TVDataset(train, model_roberta_large)
val_1 = TVDataset(val, model_roberta_large)
test_1 = TestDataset(test_df, model_roberta_large)

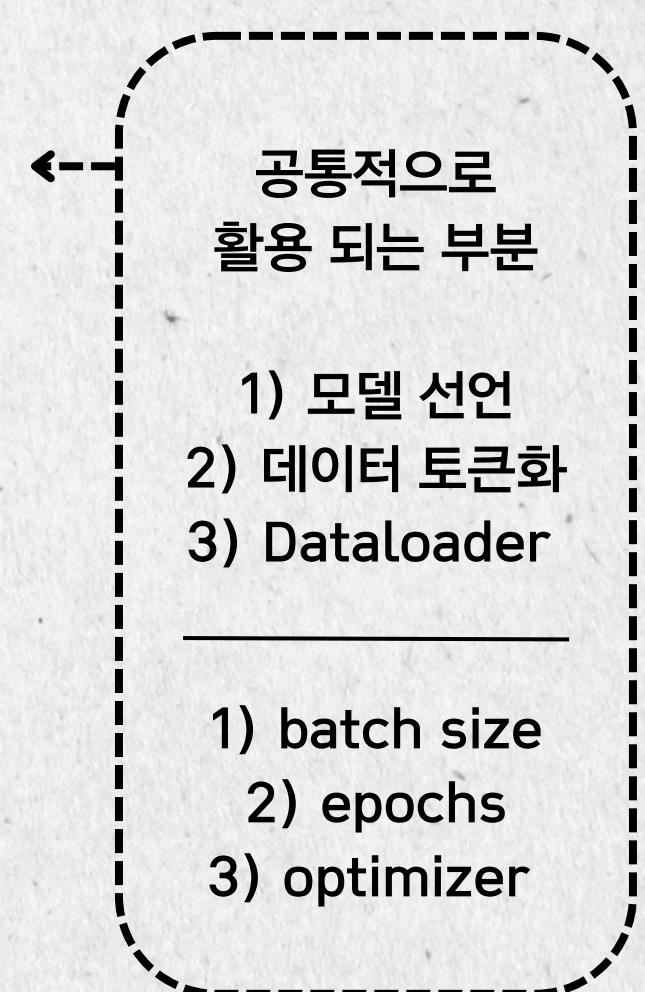
batch_size = 128

train_loader_1 = DataLoader(train_1, batch_size=batch_size, shuffle=True)
val_loader_1 = DataLoader(val_1, batch_size=batch_size, shuffle=True)
test_loader_1 = DataLoader(test_1, batch_size=batch_size, shuffle=False)

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = AutoModelForSequenceClassification.from_pretrained(model_roberta_large, num_labels=7)
model.to(device)

epochs = 3
optimizer = AdamW(model.parameters(), lr=1e-5)

```



# 04. 모델링

## 모델 1 : RoBERTa (large / small / base)

```
# train
losses = []
accuracies = []
total_loss = 0.0
correct = 0
total = 0

for i in range(epochs):
    model.train()

    for input_ids_batch, attention_masks_batch, y_batch in tqdm(train_loader_1):
        optimizer.zero_grad()
        y_batch = y_batch.to(device)
        y_pred = model(input_ids_batch.to(device), attention_mask=attention_masks_batch.to(device))[0]
        loss = F.cross_entropy(y_pred, y_batch)
        loss.backward()
        optimizer.step()

        total_loss += loss.item()

        _, predicted = torch.max(y_pred, 1)
        correct += (predicted == y_batch).sum()
        total += len(y_batch)

    losses.append(total_loss)
    accuracies.append(correct.float() / total)
    print("Train Loss:", total_loss / total, "Accuracy:", correct.float() / total)
```

```
# validation
model.eval()

pred = []
correct = 0
total = 0

for input_ids_batch, attention_masks_batch, y_batch in tqdm(val_loader_1):
    y_batch = y_batch.to(device)
    y_pred = model(input_ids_batch.to(device), attention_mask=attention_masks_batch.to(device))[0]
    _, predicted = torch.max(y_pred, 1)
    pred.append(predicted)
    correct += (predicted == y_batch).sum()
    total += len(y_batch)

print("val accuracy:", correct.float() / total)
```

```
# test
model.eval()

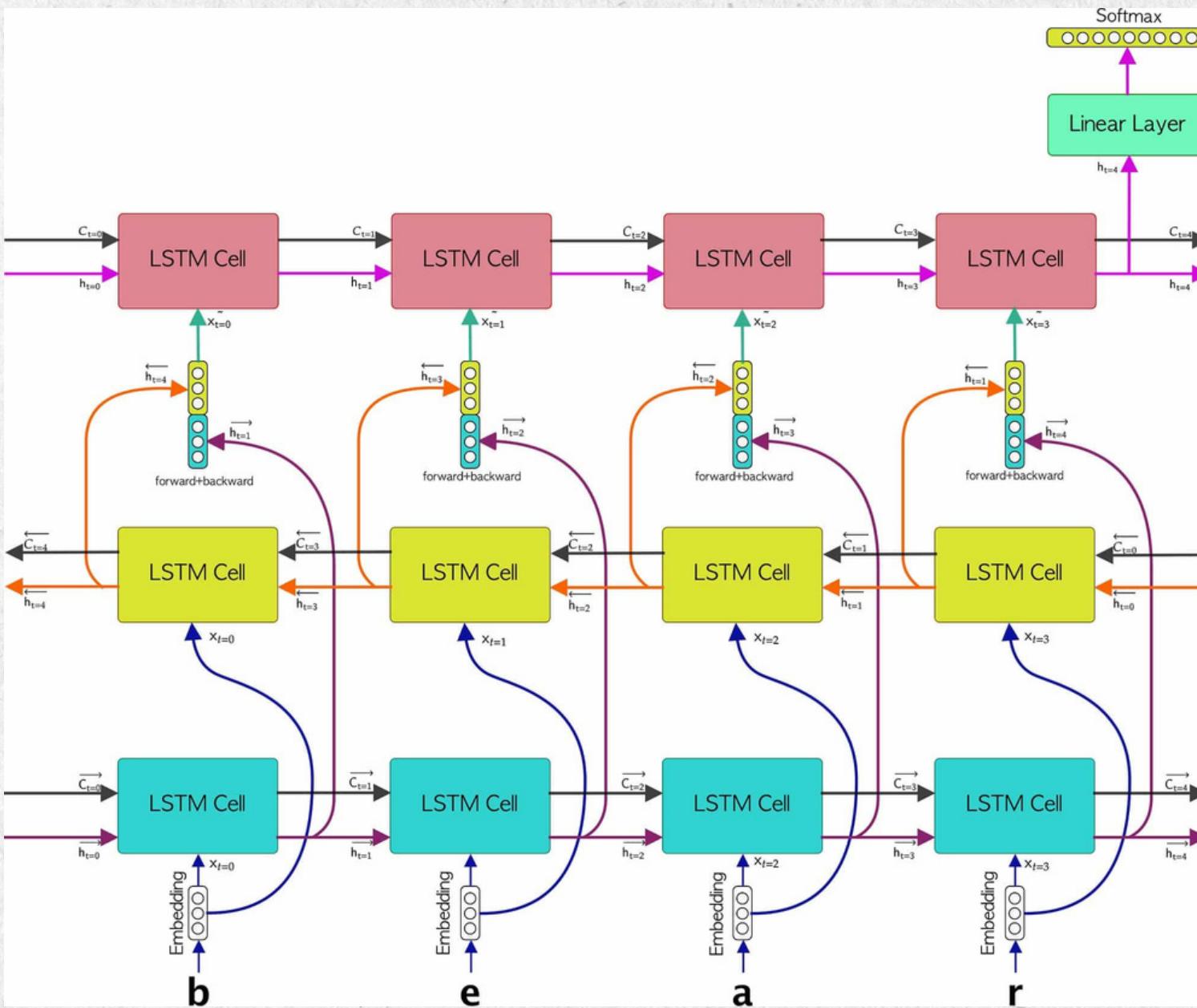
pred = []

for input_ids_batch, attention_masks_batch in tqdm(test_loader_1):
    y_pred = model(input_ids_batch.to(device), attention_mask=attention_masks_batch.to(device))[0]
    _, predicted = torch.max(y_pred, 1)
    pred.extend(predicted.tolist())
```

large / small / base 모델에 대하여 각각  
train ~ validation ~ test 과정 학습

# 04. 모델링

## 모델 2 : 양방향 LSTM



: 단방향 LSTM과 달리, 순방향 & 역방향 두 개의 LSTM 셀을 사용하는 LSTM 모델

- 대응하는 단어의 주변 정보를 균형있게 담을 수 있음
- 기존의 LSTM 계층에 역방향으로 처리하는 LSTM 계층을 추가하고, 최종 은닉상태는 두 LSTM 계층의 은닉상태를 연결한 벡터를 출력
- 출력값에 대한 손실을 최소화하는 과정에서 모든 파라미터가 동시에 학습되는 종단간 학습이 가능함

~ 일반적으로 양방향 LSTM을 사용할 경우 시퀀스 데이터에서 더 많은 정보를 추출할 수 있기 때문에 성능이 더 좋게 나타난다

## 04. 모델링

### 모델 2 : 양방향 LSTM

```

1 max_len = 15
2 vocab_size = 26124
3 embedding_dim = 128 # vocab_size보다 훨씬 작게 설정
4
5 model = Sequential()
6 model.add(Embedding(vocab_size, embedding_dim, input_length = max_len, trainable = True))
7 model.add(Bidirectional(LSTM(64, dropout = 0.2, recurrent_dropout = 0.2, return_sequences = True)))
8 model.add(Bidirectional(LSTM(64, dropout = 0.2, recurrent_dropout = 0.2, return_sequences = True)))
9 model.add(Bidirectional(LSTM(64, dropout = 0.2, recurrent_dropout = 0.2)))
10 model.add(Dense(7, activation = "softmax"))
11

```

```

1 es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=4)
2 mc = ModelCheckpoint('best_model.h5', monitor='val_acc', mode='max', verbose=1, save_best_only=True)
3
4 X_train, X_valid, y_train, y_valid = train_test_split(X_train, y_train, test_size = 0.3, random_state = 1000)
5 model.compile(loss= SparseCategoricalCrossentropy(), optimizer='adam', metrics=['acc'])
6 history = model.fit(X_train, y_train,
7                      batch_size=128, epochs=30,
8                      callbacks=[es, mc], validation_data=(X_valid, y_valid), validation_split = 0.3)

```

## 05. 결과

### <RoBERTa - large / small / base>

Large			
public : 0.8486	private : 0.8142		
small			
public : 0.8460	private : 0.8107		
base			
public : 0.8486	private : 0.8114		

### 양방향 LSTM

양방향 LSTM			
public : 0.7813	private : 0.7477		

# Thank you!

---

: 월간 데이콘 뉴스 토픽 분류 AI 경진대회

2023 SS KUBIG CONTEST

Team 1

강동현 (산업경영공학부 19)  
김나연 (통계학과 19)