multicampus

1조 인정하는지홍 ?

NEWS

토픽 분류 서비스

K-Digital Training 서비스 산업 데이터를 활용한 빅데이터 분석 4회차 D반

작성일 - 2022.04.19.(화)

고인정 김홍진 지호 한동권

목차

수행 순서, 수행도구, 데이터

데이터 수집, 전처리, 모델링,
시각화 및 탐색, AWS 서비스

Q&A

1. 프로젝트 주제 및 개요

주제

뉴스 헤드라인을 활용한
뉴스 토픽 분류 서비스

1. 프로젝트 주제 및 개요

핵심 기능

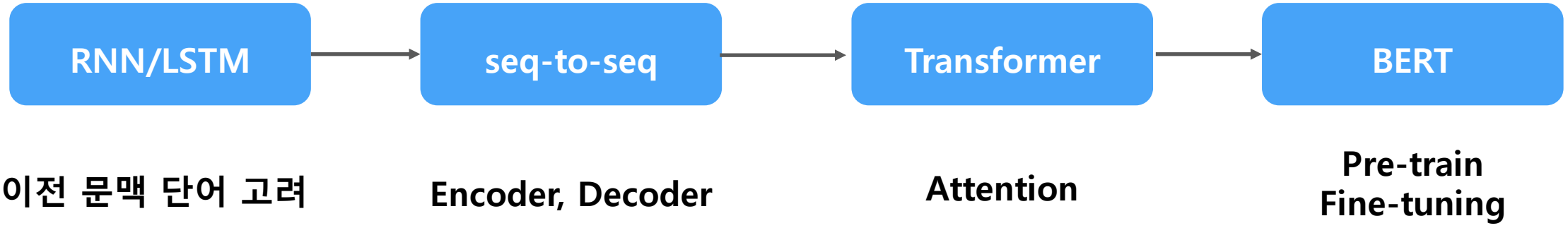
1. 입력 데이터 언어 인식
2. Roberta 모델을 활용하여 토픽 분류

2. 프로젝트 수행 방향

수행 순서

1. 자연어 데이터를 활용한 뉴스 주제 분류 기능 구현
2. 한국어 뉴스 토픽 분류 : **RoBERTa-base 모델**
3. 영어 뉴스 토픽 분류 : **DistilRoBERTa-base 모델**
4. 한국어와 영어를 구별하여 각 언어에 적합한 모델에 적용.
5. 데이터를 크롤링하고, 모델에 적용하여 확인.
6. AWS 서버에서 서비스 제공

NLP Deep Learning Model의 역사



참고 사이트 : https://sungalex.github.io/ai/ai&qa/자연어처리/2020/05/18/Study6-NLP-DeepLearning-RNN-BERT.html

2. 프로젝트 수행 방향

🛠수행 도구

구분		항목	적용내역
S/W 개발환경	데이터 분석, 모델링	Google Colab	Google과 Kaggle에서 지원하는 GPU를 사용, Python 기반으로 Tensorflow 라이브러리를 활용하여 데이터 분석과 모델링.
		Kaggle	
H/W 구성장비	서버	AWS EC2	AWS에서 지원하는 GPU를 사용, Jupyter notebook 개발환경에서 Python 기반으로 Tensorflow 라이브러리를 활용하여 데이터 분석과 모델링.

2. 프로젝트 수행 방향

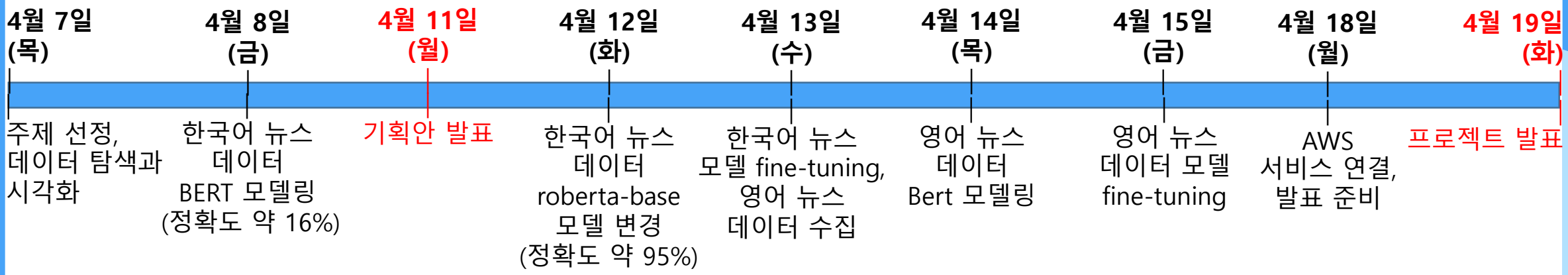
데이터

- [데이콘] 뉴스 토픽 분류 AI 경진대회 제공 데이터 :
  YNAT(주제 분류를 위한 연합 뉴스 헤드라인) 데이터 세트
- 다음 뉴스 크롤링
- [Kaggle] Topic Labeled News Dataset :
  HEALTH 데이터를 제외한 모든 데이터를 5000개씩
- [Kaggle] : News Category Dataset :
  정치 데이터만 추가

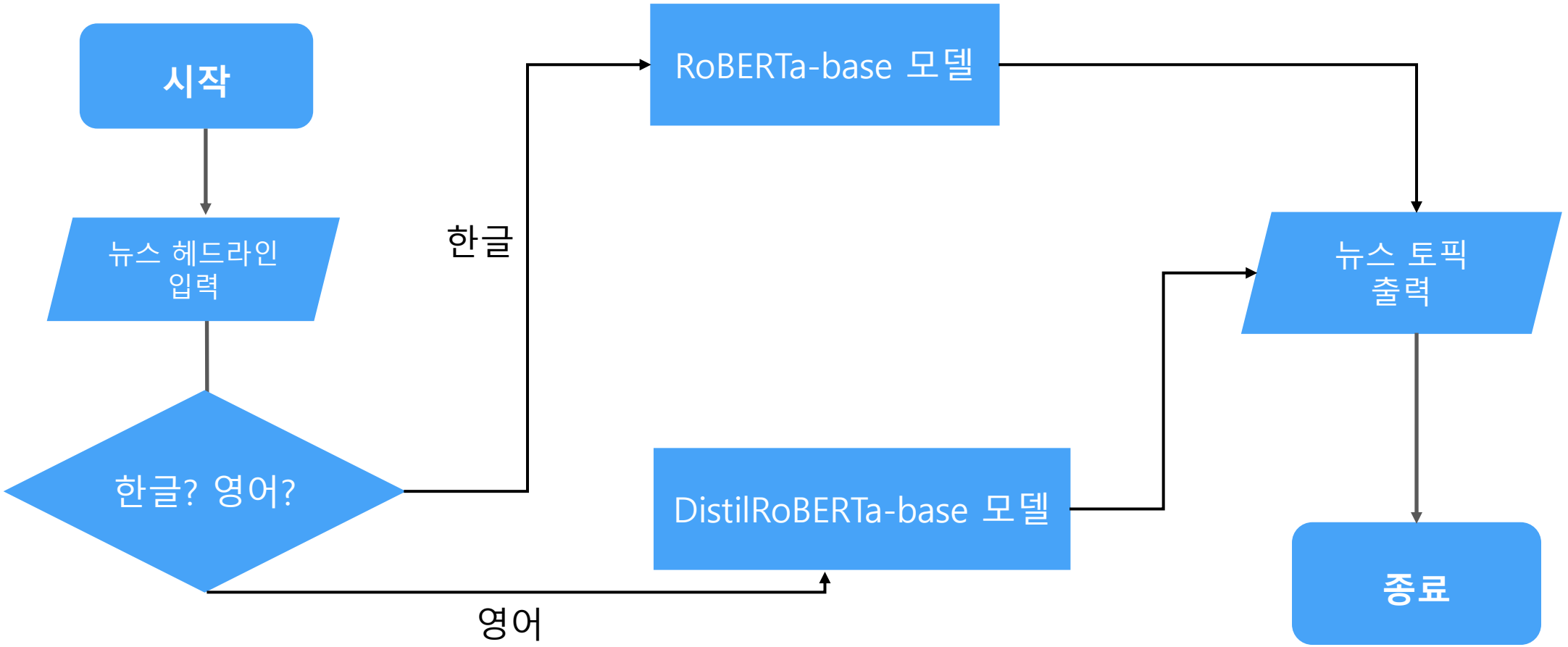
3. 프로젝트 조직 구성원 및 역할

조장	조원	조원	조원
한동권	고인정	김홍진	지호
데이터 분석, 시각화 한국어 데이터 전처리 영어 데이터 수집, 크롤링 영어 모델 구축, 파라미터 조절 프로젝트 발표	데이터 분석, 시각화 한국어 데이터 전처리 한국어 모델 구축, 파라미터 조절 영어 모델 파라미터 조절 PPT 제작 프로젝트 발표	데이터 분석, 시각화 한국어 데이터 전처리 영어 데이터 전처리, 정제 한국어 모델 구축, 파라미터 조절 AWS 서비스	데이터 분석, 시각화 영어 데이터 수집, 정제 영어 모델 구축, 파라미터 조절 기획안 발표

4. 프로젝트 일정


4월 7일 (목)
주제 선정, 데이터 탐색과 시각화
4월 8일 (금)
한국어 뉴스 데이터 BERT 모델링 (정확도 약 16%)
4월 11일 (월)
기획안 발표
4월 12일 (화)
한국어 뉴스 데이터 roberta-base 모델 변경 (정확도 약 95%)
4월 13일 (수)
한국어 뉴스 모델 fine-tuning, 영어 뉴스 데이터 수집
4월 14일 (목)
영어 뉴스 데이터 Bert 모델링
4월 15일 (금)
영어 뉴스 데이터 모델 fine-tuning
4월 18일 (월)
AWS 서비스 연결, 발표 준비
4월 19일 (화)
프로젝트 발표

전체 흐름도


시작
뉴스 헤드라인 입력
한글? 영어?
한글
영어
RoBERTa-base 모델
DistilRoBERTa-base 모델
뉴스 토픽 출력
종료

프로그램 구현

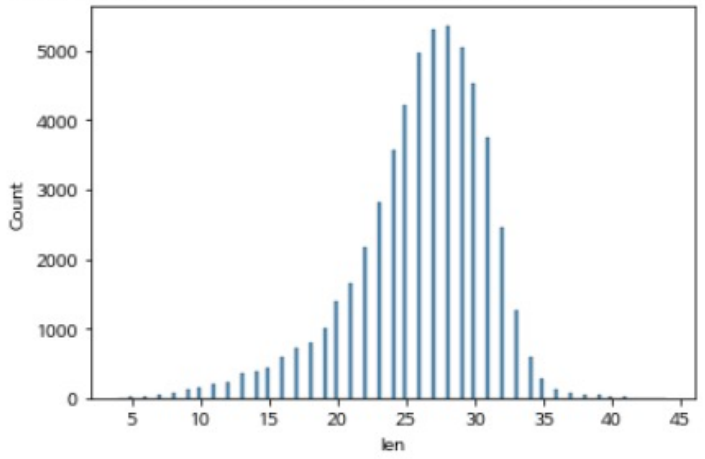
토픽 분류 서비스

K-Digital Training 서비스 산업 데이터를 활용한 빅데이터 분석 4회차 D반

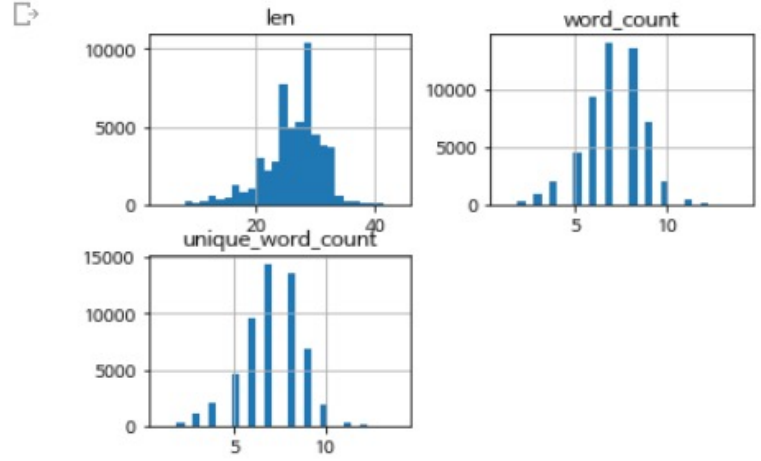
5. 프로젝트 수행 한국어 데이터 시각화 및 탐색

```
# 서브플롯을 통해 "len", "word_count", "unique_word_count"의 histplot을 시각화 한다
sns.histplot(data=df, x="len")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f542a4c3d10>
```



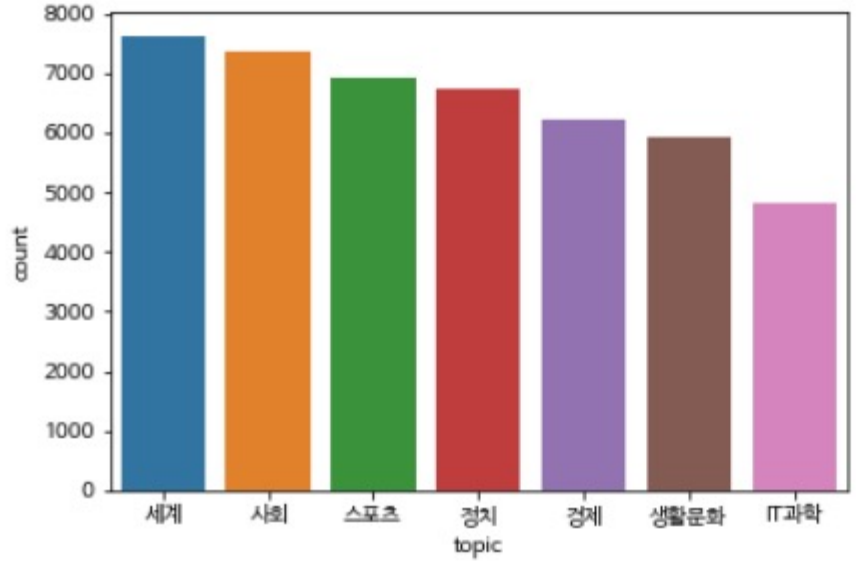

```
# 확인해본 결과 특별히 눈에 띄는 값은 없다.
df[["len", "word_count", "unique_word_count"]].hist(bins=30)
plt.show()
```



5. 프로젝트 수행 한국어 데이터 시각화 및 탐색

```
# train2 로 빈도수를 구했지만 test 데이터는 topic이 결측치라 포함되지 않는다.
# order value_counts 기준으로 정렬을 해준다.
sns.countplot(data=df, x='topic', order=df["topic"].value_counts().index)
```

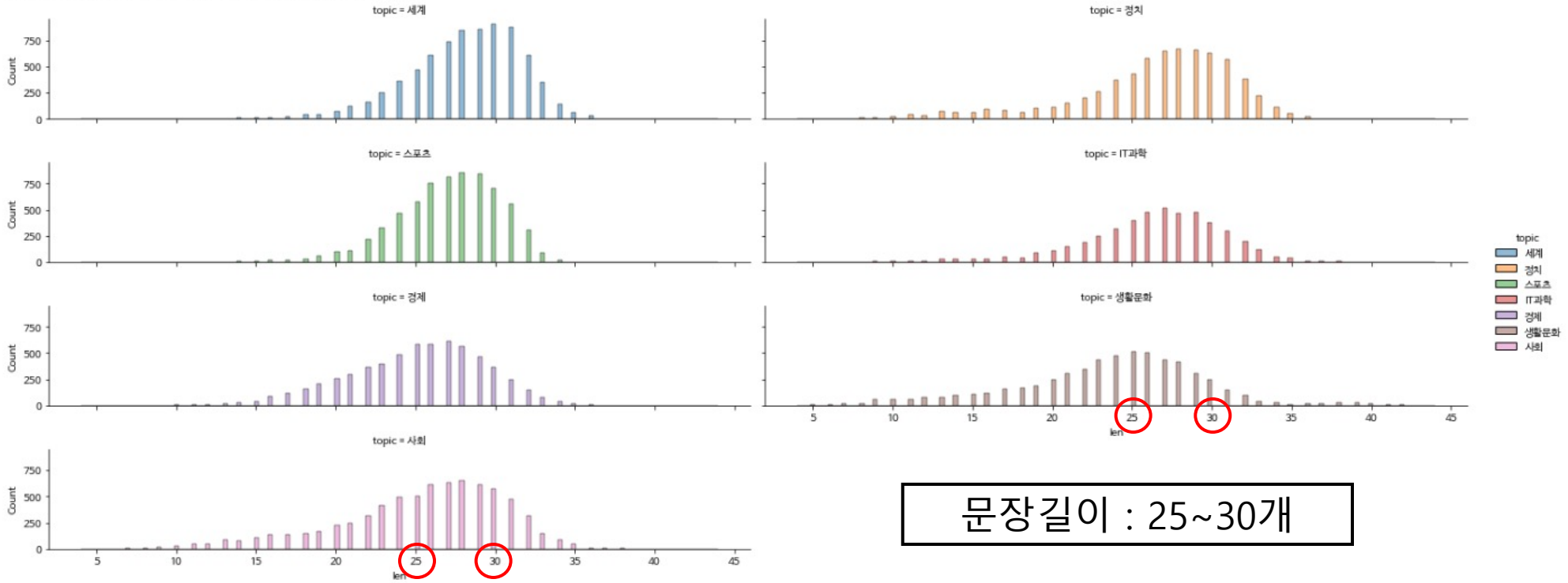
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f542b710450>
```



5. 프로젝트 수행 한국어 데이터 시각화 및 탐색

```
# 문장길이
# displot 으로 topic 별 "len" 의 histplot 시각화 하기
# data=df, x="len", kind="hist", hue="topic", col="topic", col_wrap=2, aspect=5, height=2
sns.displot(data=df, x="len",hue="topic", col="topic", col_wrap=2, aspect=5, height=2)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f542a4df8d0>
```

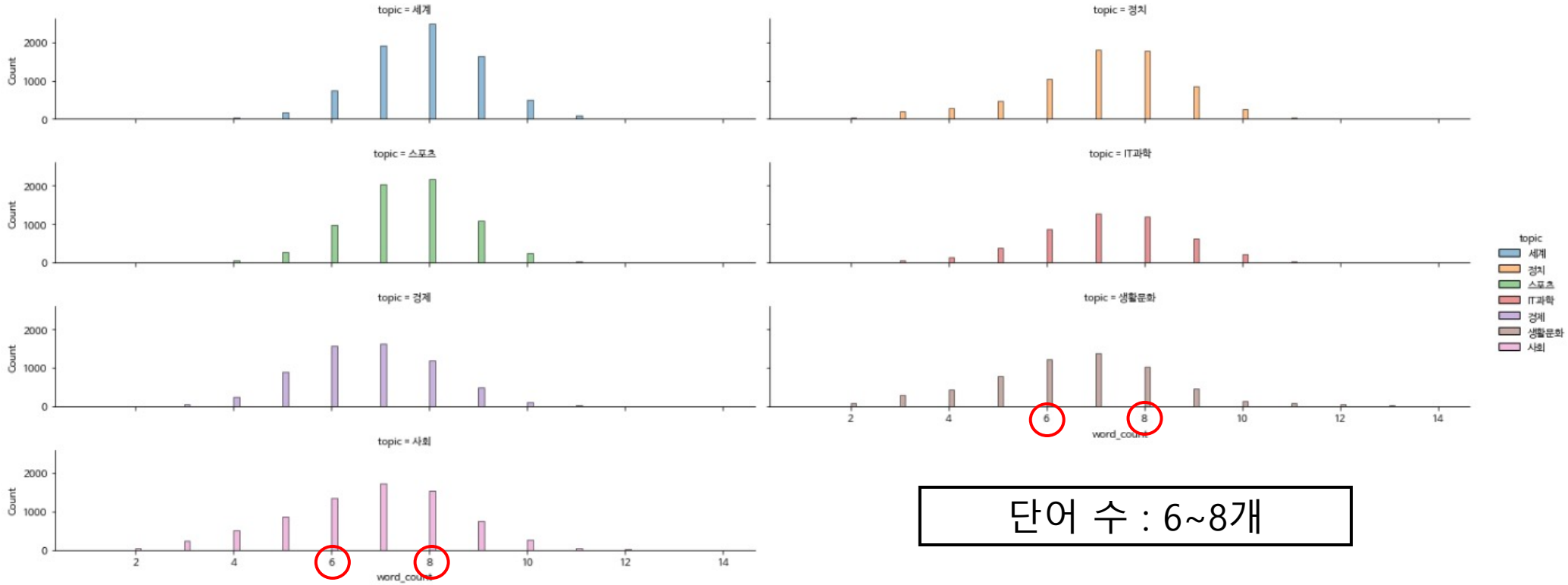


문장길이 : 25~30개

5. 프로젝트 수행 한국어 데이터 시각화 및 탐색

```
# 단어 수
# displot 으로 topic 별 "word_count"의 histplot 시각화 하기
sns.displot(data=df, x="word_count",hue="topic",col="topic", col_wrap=2, height=2, aspect=5)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f5426970790>
```

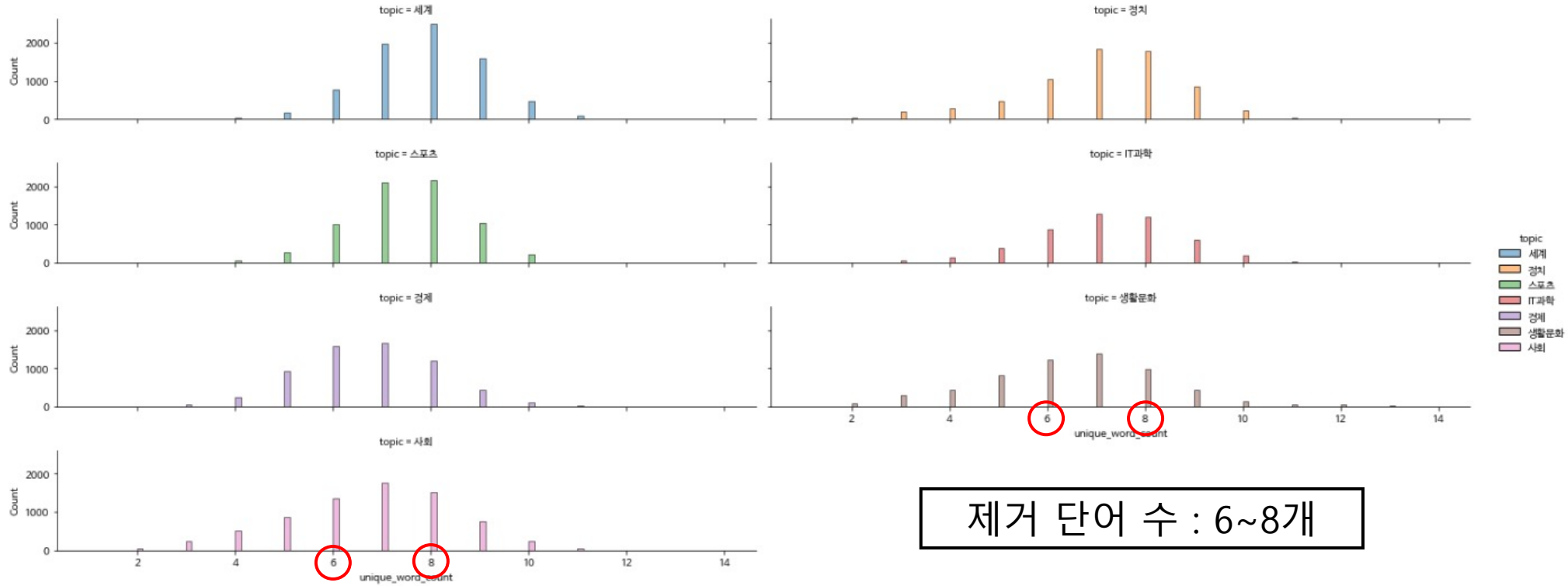


단어 수 : 6~8개

5. 프로젝트 수행 한국어 데이터 시각화 및 탐색

```
# 중복 제거 단어 수
# displot 으로 topic 별 "unique_word_count" 의 histplot 시각화 하기
sns.displot(data=df, x="unique_word_count", hue="topic", col="topic", col_wrap=2, height=2, aspect=5)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f54268cb590>
```



제거 단어 수 : 6~8개

5. 프로젝트 수행 한국어 데이터 시각화 및 탐색

```
# content를 위해서 만든 함수인 display_word_cloud로 워드클라우드를 시각화 한다.
display_word_cloud(content)
```

5. 프로젝트 수행 한국어 데이터 시각화 및 탐색

```
# content를 위해서 만든 함수인 display_word_cloud로 워드클라우드를 시각화 한다.
display_word_cloud(content)
```

5. 프로젝트 수행

한국어 데이터 모델링

```
model_checkpoint="yobi/klue-roberta-base-ynat"
batch_size = 32
task = "ynat"
num_epochs = 5
learningRate = 2e-5
weight_decay = 0.01
seed_value = 42
num_labels = 7
```

```
metric = load_metric("accuracy")
```

Weight decay 가중치 감쇠 : Overfitting 방지

Seed value : 매번 동일한 난수값을
생성하기 위해 최초로 정해놓은 값.

5. 프로젝트 수행

한국어 데이터 모델링

```
tokenizer = AutoTokenizer.from_pretrained(model_checkpoint)
```

```
def preprocess_function(examples):
    return tokenizer(
        examples['title'],
        padding="max_length",
        max_length = 40,
        truncation=True
    )
```

```
model = AutoModelForSequenceClassification.from_pretrained(model_checkpoint, num_labels=num_labels)
```

5. 프로젝트 수행

한국어 데이터 모델링

```
def compute_metrics(eval_pred):
    predictions, labels = eval_pred
    print(predictions)
    predictions = np.argmax(predictions, axis=1)
    return metric.compute(predictions=predictions, references=labels)
```

Predicitons : 인덱스별 예측 확률값
Labels : 토픽 인덱스

5. 프로젝트 수행

한국어 데이터 모델링

```
metric_name = "accuracy"


args = TrainingArguments(
    #"test-nli",
    "test-ynat",
    evaluation_strategy="epoch",
    save_strategy = "epoch",
    learning_rate= learningRate,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=num_epochs,
    weight_decay=weight_decay,
    #seed = seed_value,
    load_best_model_at_end=True,
    metric_for_best_model= metric_name
)
```

5. 프로젝트 수행

한국어 데이터 모델링

```
trainer = Trainer(
    model,
    args,
    train_dataset=encoded_dataset_train,
    eval_dataset=encoded_dataset_validation,
    tokenizer=tokenizer,
    compute_metrics=compute_metrics,
)
```

```
trainer.train()
```

```
trainer.evaluate()
```

```
trainer.evaluate()
[286/286 00:21]
[[-1.2603986  -1.3839709   0.86399907 ...  0.65533775 -1.5621202
   6.215123  ]
 [-0.5741025  -0.7315076   5.353249   ... -1.4809321  -3.0208411
   0.01927516]
 [-1.0406541   5.6695504   0.28884774 ... -1.1129158  -2.628502
  -2.4857829 ]
 ...
 [-0.62745875 -0.940294    5.3889656  ... -1.3820916  -2.9110112
  -0.07743073]
 [-0.72990614 -1.1181817   5.4037786  ... -1.2909156  -2.768382
  -0.21478003]
 [-1.274492   -1.3770021   0.8682788  ...  0.65586066 -1.5595403
   6.2150493 ]]
{'epoch': 5.0,
 'eval_accuracy': 0.9462271383200087,
 'eval_loss': 0.2663747668266296,
 'eval_runtime': 21.8883,
 'eval_samples_per_second': 417.164,
 'eval_steps_per_second': 13.066}
```

5. 프로젝트 수행

한국어 데이터 모델링

Train 데이터

```
dataset_train[0]['title']
```

```
'女배구 흥국생명도로공사 개막전 테일러 매치 앞두고 신경전'
```

```
classifier(dataset_train[0]['title'])
```

```
[[{'label': 'LABEL_0', 'score': 0.00071470439434051 51},
  {'label': 'LABEL_1', 'score': 0.00063509534811601 04},
  {'label': 'LABEL_2', 'score': 0.00154820375610142 95},
  {'label': 'LABEL_3', 'score': 0.00054047227604314 68},
  {'label': 'LABEL_4', 'score': 0.00929854065179824 8},
  {'label': 'LABEL_5', 'score': 0.9860885143280029},
  {'label': 'LABEL_6', 'score': 0.00117450987454503 77}]]
```

```
print(dataset_train[0]['label'])
```

```
5
```

Test 데이터

```
classifier(dataset_validation[0]['title'])
```

```
[[{'label': 'LABEL_0', 'score': 0.00038322218460962176},
  {'label': 'LABEL_1', 'score': 0.00041732590761967003},
  {'label': 'LABEL_2', 'score': 0.006987015251070261},
  {'label': 'LABEL_3', 'score': 0.00026505152345634997},
  {'label': 'LABEL_4', 'score': 0.0034160108771175146},
  {'label': 'LABEL_5', 'score': 0.0005788246635347605},
  {'label': 'LABEL_6', 'score': 0.987952470779419}]]
```

```
datasets['test']['title'][0]
```

```
'대통령개헌안 초점 대통령권한 내려놓기…국회 · 총리 힘 세져'
```

```
print(dataset_validation[0]['label'])
```

```
6
```

5. 프로젝트 수행

한국어 데이터 전처리

```
import re

def preprocessing(text):
  # 개행문자 제거
  text = re.sub('\\\\n', ' ', text)

  # 특수문자 제거
  # 한글, 영문, 한자만 남기고 모두 제거
  text = re.sub('[^가-힣ㄱ-ㅎㅏ-ㅣa-zA-Z一-顯]', ' ', text)

  # 중복으로 생성된 공백값 제거
  text = re.sub('[\s]+', ' ', text)

  # 영문자를 소문자로 만듦
  text = text.lower()
  return text
```

```
train['title'] = train['title'].map(preprocessing)
test['title'] = test['title'].map(preprocessing)
```

5. 프로젝트 수행

한국어 데이터 전처리

```
def remove_stopwords(text):
  tokens = text.split(' ')
  stops = ['합니다', '하는', '할', '하고', '한다',
           '그리고', '입니다', '그', '등', '이런', '및', '제', '더']
  meaningful_words = [w for w in tokens if not w in stops]
  return ' '.join(meaningful_words)
```

```
# map을 통해 불용어 일괄 제거
train['title'] = train['title'].map(remove_stopwords)
test['title'] = test['title'].map(remove_stopwords)
```

5. 프로젝트 수행

한국어 데이터 모델링

BERT 모델
⇒ 낮은 정확도(18%) :
데이터셋을 학습시켜 모델 생성시 성능이 매우 낮게 나오고 있는 원인 확인 필요.

모델이 적합하지 않아, 학습이 제대로 이루어지지 않음.
⇒ **klue-roberta-base 모델**로 변경
⇒ 정확도(95%) :
전처리시 정확도가 변경되는 것을 확인함.

전처리x - acc: 0.9462, loss: 0.2663
한자 유 불용어 제거 유 - acc: 0.9372, loss: 0.2992
한자 무, 특수문자 유, 불용어 제거 유 - acc:0.9305, loss:0.3213
한자 무, 특수문자 유, 불용어 제거 무 - acc:0.9201, loss: 0.3524
한자 유, 특수문자 무, 불용어 제거 무 - acc:0.9364 ,loss: 0.3059
한자 무, 특수문자 무, 불용어 제거 유 - acc: 0.9208, loss: 0.3424

전처리를 하지 않았을 때,
정확도가 제일 높게 나왔음.

5. 프로젝트 수행

크롤링 데이터 모델 테스트

```
trainer.train()
```

[7135/7135 27:11, Epoch 5/5]

Epoch	Training Loss	Validation Loss	Accuracy
1	0.270100	1.940000	0.590984
2	0.222800	2.046813	0.581967
3	0.196400	1.933892	0.612295
4	0.173400	2.094243	0.590984
5	0.160000	2.085848	0.601639

```
trainer.evaluate()
```

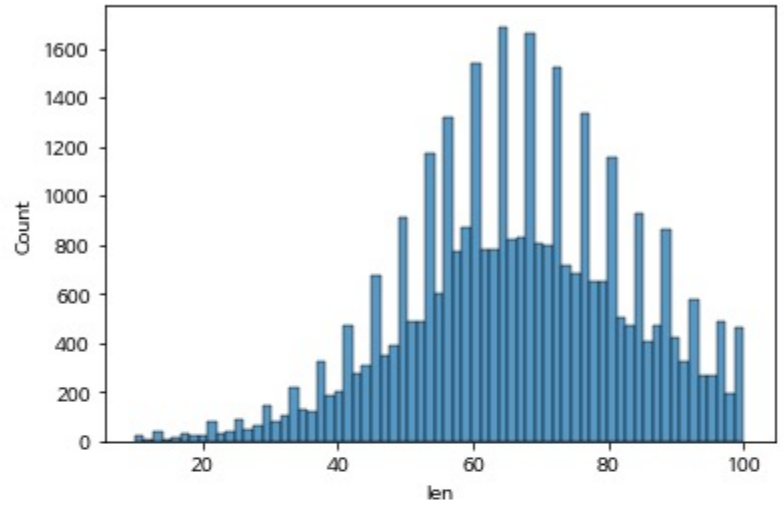
```
{'eval_loss': 1.9338918924331665,
 'eval_accuracy': 0.6122950819672132,
 'eval_runtime': 2.7607,
 'eval_samples_per_second': 441.924,
 'eval_steps_per_second': 14.127,
 'epoch': 5.0}
```

5. 프로젝트 수행

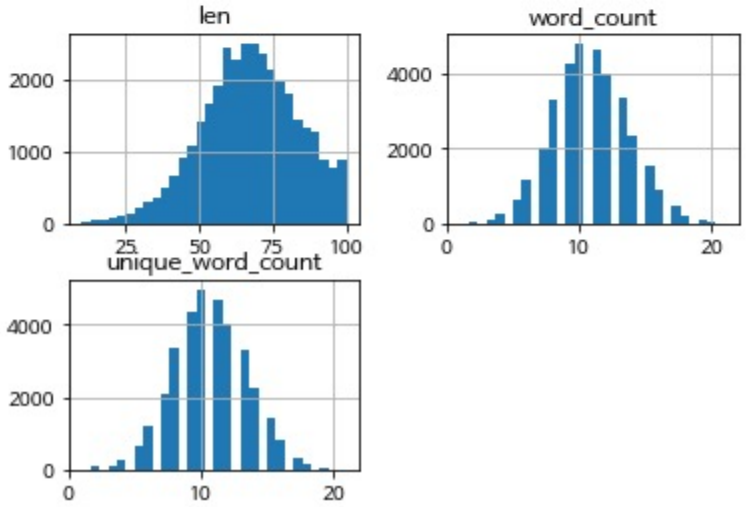
영어 데이터 시각화 및 탐색

```
# 서브플롯을 통해 "len", "word_count", "unique_word_count"의 histplot을 시각화 한다
sns.histplot(data=df, x="len")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc994535d90>
```




```
[18] # 확인해본 결과 특별히 눈에 띄는 값은 없다.
     df[["len", "word_count", "unique_word_count"]].hist(bins=30)
     plt.show()
```

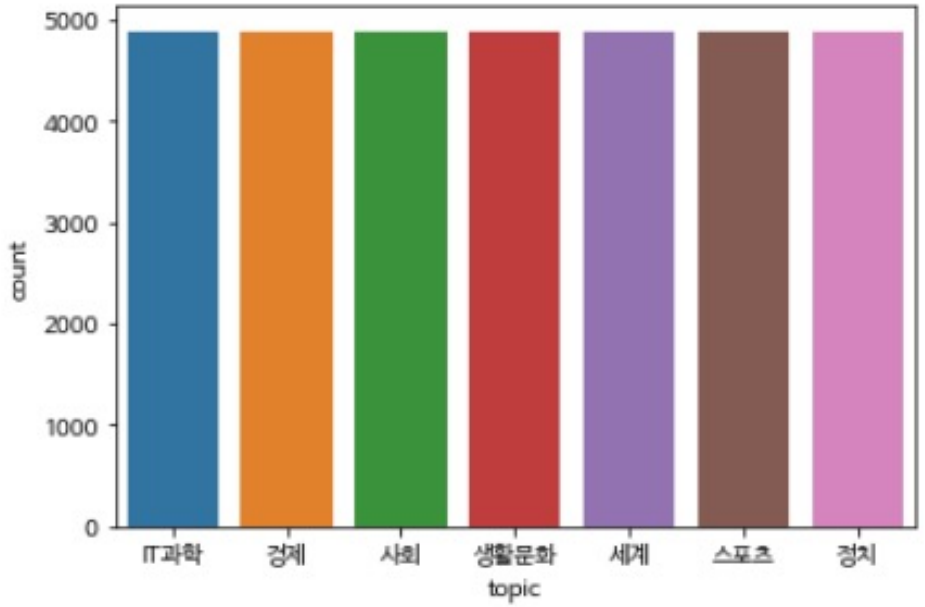


5. 프로젝트 수행

영어 데이터 시각화 및 탐색

```
# train2 로 빈도수를 구했지만 test 데이터는 topic이 결측치라 포함되지 않는다.
# order value_counts 기준으로 정렬을 해준다.
sns.countplot(data=df, x='topic', order=df["topic"].value_counts().index)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc99456bb90>
```

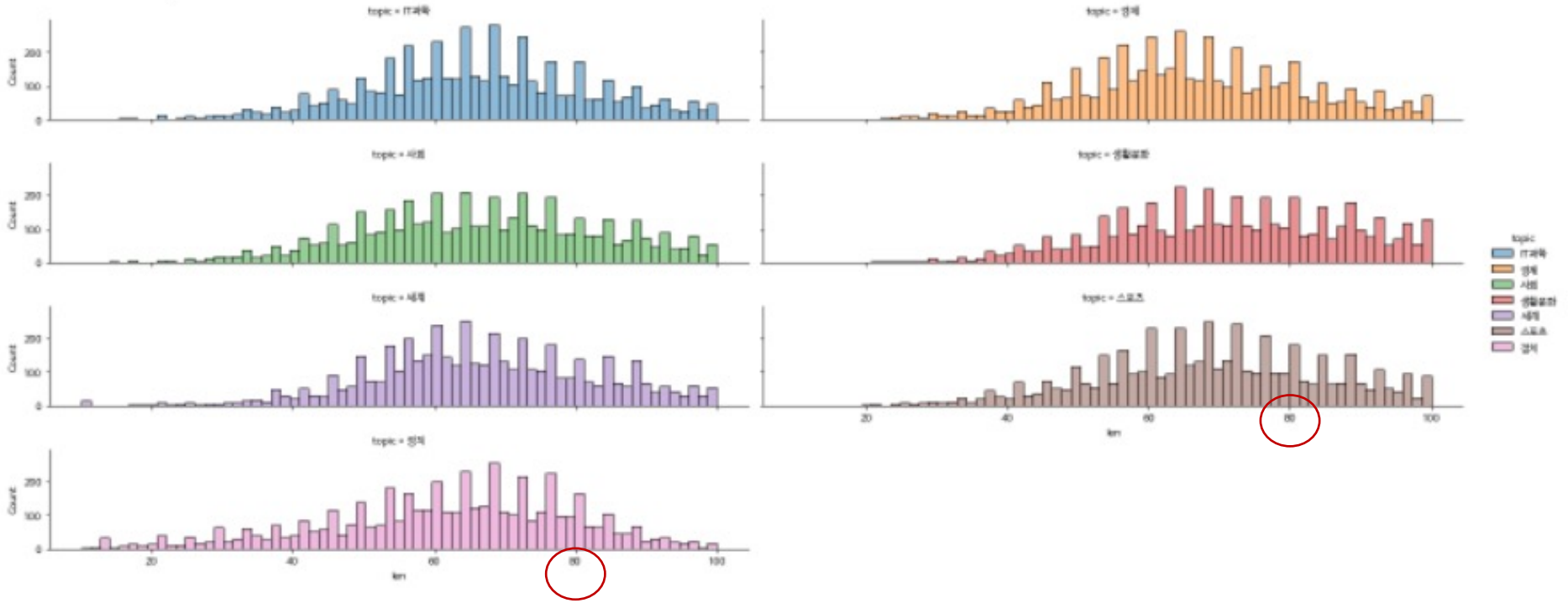


5. 프로젝트 수행

영어 데이터 시각화 및 탐색

```
[19] # 문장길이
     # displot 으로 topic 별 "len" 의 histplot 시각화 하기
     # data=df, x="len", kind="hist", hue="topic", col="topic", col_wrap=2, aspect=5, height=2
     sns.displot(data=df, x="len",hue="topic", col="topic", col_wrap=2, aspect=5, height=2)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc993a8dd10>
```

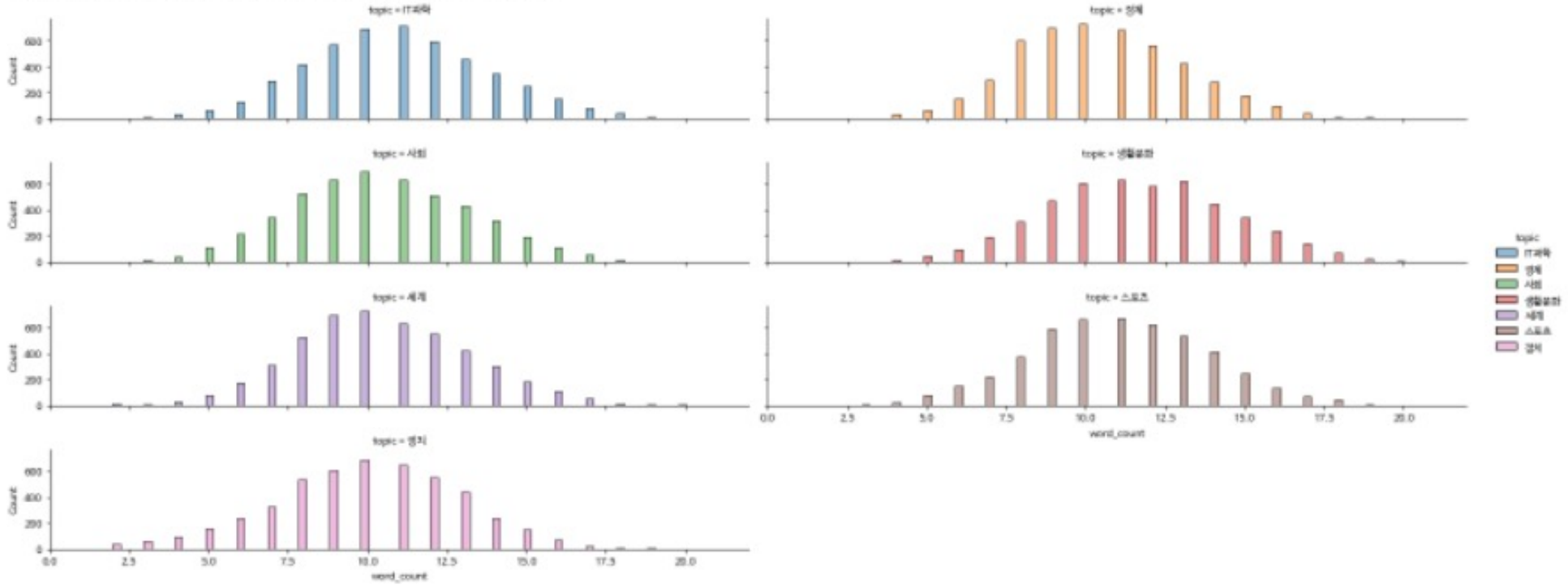


5. 프로젝트 수행

영어 데이터 시각화 및 탐색

```
# 단어 수
# displot 으로 topic 별 "word_count"의 histplot 시각화 하기
sns.displot(data=df, x="word_count",hue="topic",col="topic", col_wrap=2, height=2, aspect=5)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc990cd3090>
```

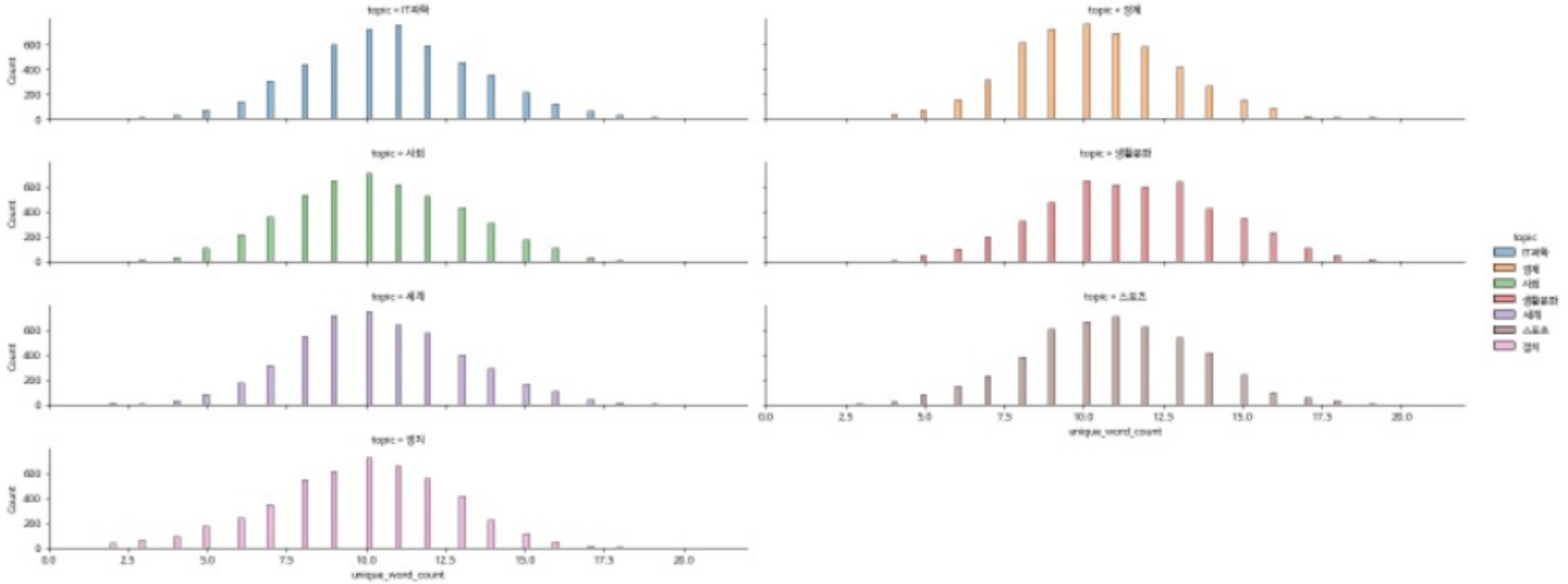


5. 프로젝트 수행

영어 데이터 시각화 및 탐색

```
# 중복 제거 단어 수
# displot 으로 topic 별 "unique_word_count" 의 histplot 시각화 하기
sns.displot(data=df, x="unique_word_count", hue="topic", col="topic", col_wrap=2, height=2, aspect=5)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc990cd3b10>
```



5. 프로젝트 수행

영어 데이터 시각화 및 탐색

```
[25]  # content를 위해서 만든 함수인 display_word_cloud로 워드클라우드를 시각화 한다.
      display_word_cloud(content)
```

5. 프로젝트 수행

영어 데이터 시각화 및 탐색

```
[30] # content를 위해서 만든 함수인 display_word_cloud로 워드클라우드를 시각화 한다.
     display_word_cloud(content)
```

5. 프로젝트 수행

영어 데이터 모델링

```
from simpletransformers.classification import MultiLabelClassificationModel

model = MultiLabelClassificationModel('roberta', 'distilroberta-base', num_labels=7,
                                      args={'train_batch_size':2, 'gradient_accumulation_steps':16,
                                     'learning_rate': 4e-5, 'num_train_epochs': 3, 'max_seq_length': 100})
```

Downloading: 100% 480/480 [00:00<00:00, 14.2kB/s]

Downloading: 100% 316M/316M [00:14<00:00, 31.0MB/s]

참고 자료

Search this file...

	Name	MCC
1	distilbert-base-uncased	0.8799310655077069
2	bert-base-cased	0.9054343936832274
3	bert-base-multilingual-cased	0.9058351469080927
4	roberta-base	0.9105405153604827
5	xlnet-base-cased	0.9133994215793382
6	distilbert-base-multilingual-cased	0.9163505446098033
7	distilroberta-base	0.9205519245675056

mcc.csv hosted with ❤ by GitHub view raw

5. 프로젝트 수행

영어 데이터 모델링

정치를 제외한 나머지 데이터

	len_title
count	98739.000000
mean	71.628870
std	22.962831
min	4.000000
25%	58.000000
50%	69.000000
75%	83.000000
max	314.000000

추가한 정치 데이터

	title
count	4965.000000
mean	62.918026
std	19.500086
min	5.000000
25%	52.000000
50%	64.000000
75%	75.000000
max	260.000000

5. 프로젝트 수행

영어 데이터 모델링

```
model.train_model(train_df)
```

Epoch 3 of 3: 100% 3/3 [24:18<00:00, 484.57s/it]

Epochs 0/3. Running Loss: 0.0446: 100% 13678/13678 [08:09<00:00, 32.63it/s]

Epochs 1/3. Running Loss: 0.0198: 100% 13678/13678 [08:06<00:00, 31.97it/s]

Epochs 2/3. Running Loss: 0.0087: 100% 13678/13678 [07:55<00:00, 24.37it/s]

5. 프로젝트 수행

영어 데이터 모델링

```
result, model_outputs, wrong_predictions = model.eval_model(eval_df)
```

Running Evaluation: 100% 855/855 [00:15<00:00, 57.42it/s]

```
result, model_outputs # 정확도, eval_loss 출력
```

```
{'LRAP': 0.9133168313384686 'eval_loss': 0.11891357672798354}
```

5. 프로젝트 수행

영어 데이터 모델링

```
predictions, raw_outputs = model.predict(["Sam was a Wizard"]) # 테스트용
```

```
100% 1/1 [00:00<00:00, 4.03it/s]
100% 1/1 [00:00<00:00, 16.94it/s]
```

```
predictions, raw_outputs
```

```
[[0, 0, 0, 1, 0, 0, 0]]
```

5. 프로젝트 수행

영어 데이터 모델링

```
print(data2['title'][11111])
print(data2['labels'][11111])
```

```
Crisis looming in childcare sector as operators fear for their survival
2
```

```
predictions, raw_outputs = model.predict([data2['title'][11111]])
```

```
100% 1/1 [00:00<00:00, 3.83it/s]
100% 1/1 [00:00<00:00, 21.62it/s]
```

```
predictions, raw_outputs
```

```
([[0, 0, 1, 0, 0, 0, 0]],
```

5. 프로젝트 수행

영어 데이터 전처리

```
import re

def preprocessing(text):
  # 개행문자 제거
    text = re.sub('\\\\n', ' ', text)

    # 한글만 제거
    text = re.sub('[가-힣ㄱ-ㅎㅏ-ㅣ]', ' ', text)

  # 중복으로 생성된 공백값 제거
    text = re.sub('[\s]+', ' ', text)
    return text


data['title'] = data['title'].map(preprocessing)
```

5. 프로젝트 수행

영어 데이터 모델링

낮은 정확도(70~80%)
1. 데이터양 늘리기
    (1). 토픽이 같은, 정치 데이터 포함 : 약 3만 4천개 (en_final_dataset.csv)
    (2). 토픽이 다른, 정치 데이터 불포함, 건강 데이터 포함 : 약 9만 9천개 (en_news_with_health.csv)
    (3). 제목+내용 데이터로 모델 돌려보기
2. 모델 변경 : 정확도가 높은 모델을 골라서 비교
3. 파라미터 조정 (maxlen = 80 -> maxlen = 100)

영어 뉴스 파라미터 조절 결과

train_batch_size = 16, gradient_accumulation_steps = 16, learning_rate = 3e-5, num_train_epochs = 3
👉 0.900

train_batch_size = 2, gradient_accumulation_steps = 16, learning_rate = 4e-5, num_train_epochs = 3
👉 0.913

5. 프로젝트 수행

에러로 인해 AWS 서버를 통한 웹 서비스는 구현하지 못 했습니다.

OSError: Unable to load weights from pytorch checkpoint file for

AWS 서비스

참고 사이트 : https://github.com/tatiblockchain/python-deep-learning-chatbot

5. 프로젝트 수행

언어 구별 코드

```
def ko_en(text):
  text = re.sub('\\\\n', ' ', text)
  text = re.sub('[^가-힣ㄱ-ㅎㅏ-ㅣa-zA-Z]', ' ', text)
  text = re.sub('[\s]+', '', text)
  text = text.lower()

  for txt in text.split():
    cnt_en, cnt_ko = 0, 0
    for i in txt:
      for j in i:
        if ord('ㄱ') <= ord(j) <= ord('힣'):
          cnt_ko += 1
        elif ord('a') <= ord(j) <= ord('z'):
          cnt_en += 1

  if len(text):
    if (cnt_ko / len(text)) >= 0.2:
      return 'Korean : klue-roberta-base-ynat 모델에 적용해주세요'
    else:
      return 'English : distilroberta-base 모델에 적용해주세요'
```

```
print(ko_en('국가에 의한 infant care와 education… "직장이나 엄마냐" 양자택일 없는 사회[청년이 외친다, ESG 나와라](18)'))
```

```
Korean : klue-roberta-base-ynat 모델에 적용해주세요
```

6. 프로젝트 결과

한국어 뉴스 토픽 분류

RoBERTa-base 모델

KLUE 데이터를 학습한 모델, 데이터 전처리 안 함.

95%

영어 뉴스 토픽 분류

DistilRoBERTa-base 모델

데이터 전처리 한글만 제거.

91%

Q&A


THANK
YOU