

연구 일자	
연구 제목	
연구 목적	
연구 내용	

- 자연어 처리를 이용한 온라인 쇼핑몰 상품 설명 분류 모델
- 온라인 쇼핑몰의 상품 설명 이미지 안의 문구의 중요도 판단
- * 상품 이름, 기능 -> 중요함 – 상품 사용에 필요한 정보
- * 상품 배송, 교환 -> 중요하지 않음 – 상품 사용에 필요하지 않은 정보
- 기존의 모델을 사용할 경우 전이 학습(Transfer Learning)이 가능해야 함

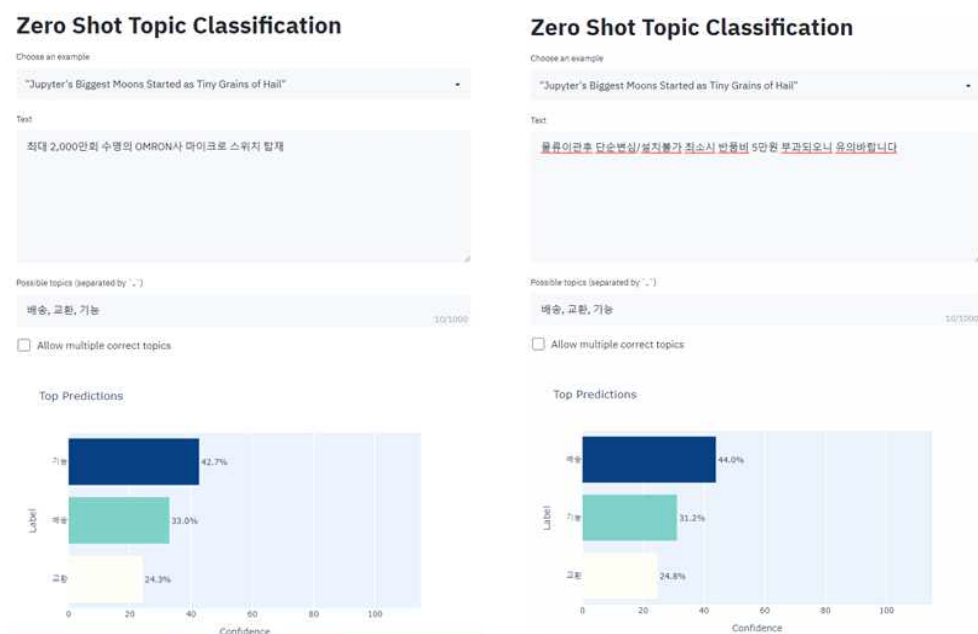
[PORORO: Platform Of neuRal mOdels for natuRal language prOcessing]

카카오브레인(Kakao Brain) kakaobrain.github.io/pororo/index.html#

- Text Classification, Sequence Tagging, Seq2Seq 등 자연어처리 기능 지원

Text Classification – Zero Shot Topic Classification

- 입력: Text(분류 대상), Topic(분류 기준)
- 출력: Text가 각 Topic에 속할 확률



live demo: 35.208.71.201:8000/

연구 일자	
연구 제목	
연구 목적	
연구 내용	PORORO Zero-shot Topic Classification Test

PORORO Zero-shot Topic Classification Test

- 테스트 방법 계획

- 1) 홈쇼핑 사이트 상 특정 상품의 상품 상세 설명 이미지 준비
 - 2) 이미지 상의 문구 문서화(수작업)
 - 3) 분류 키워드 정리
 - 3) 문서화된 문구와 분류 키워드를 PORORO Zero-shot Topic Classification에 입력하여 분류 실행
- ex)

제품 파손시 안내사항

캔 상품 등의 특성상 택배 배송과정에서 불가피하게 일부 살짝 찌그러지는 경우가 있습니다. 이점 구매시 꼭 참고 바랍니다.

☞ 상품의 단순 찌그러짐 및 스크래치 발생시 반품 및 교환이 어려울 수 있습니다.

📷 심한 파손 발생시, 파손된 제품 사진 촬영바랍니다.

- 1) ☎ 02-599-6966 연락주시면 안내후 파손품처리에 도움 드리도록 하겠습니다.

2)

제품 파손시 안내사항

캔 상품 등의 특성상 택배 배송과정에서 불가피하게 일부 살짝 찌그러지는 경우가 있습니다. 이점 구매시 꼭 참고 바랍니다.

상품의 단순 찌그러짐 및 스크래치 발생시 반품 및 교환이 어려울 수 있습니다.

심한 파손 발생시, 파손된 제품 사진 촬영바랍니다.

02-599-6966

연락주시면 안내후 파손품처리에 도움 드리도록 하겠습니다.

- 3) 분류 키워드: 이름, 기능, 외형, 수상, 배송, 교환

- 테스트 특이사항:

특정 제품군에서만 좋은 결과를 얻을 수 있기 때문에 다양한 종류의 상품 테스트 식품, 가전제품, 의류 등으로 계획

연구 일자	
연구 제목	
연구 목적	
연구 내용	PORORO Zero-shot Topic Classification Test

코드- github.com/kakaobrain/pororo

1) tests/test_zero_shot_classification.py

```
1 import unittest
2 from pororo import Pororo
3
4 class PororoZSLTester(unittest.TestCase):
5     def test_modules(self):
6         zsl = Pororo(task="zero-topic", lang="ko")
7         zsl_res = zsl(
8             """제품 파손시 안내사항
9             캔 상판 등의 특장상 학배 배송과정에서 물기피해에 일부 실측
10             피그리지는 경우가 있습니다. 이점 구매시 꼭 참고 바랍니다.
11             상품의 단층 피그리짐 및 스크래치 발생시 반품 및 교환이 어려울 수 있습니다.
12             심한 파손 발생시, 파손된 제품 사진 촬영바랍니다.
13             02-599-6900
14             연락주시면 안내후 파손품처리에 도움 드리도록 하겠습니다.""",
15             ["아름", "기능", "외형", "구성", "수상", "배송", "교환", "주의사항"],
16         )
17         self.assertIsInstance(zsl_res, dict)
18         print(zsl_res)
19
20 if __name__ == "__main__":
21     unittest.main()
```

- 해당 코드에 Text(분류 대상)과 Topic(분류 기준)을 인자로 입력
- 인자의 정보는 Pororo의 zero-topic(PororoZeroShot에서 분류를 수행

2) pororo/tasks/zero_shot_classification.py

```
8 class PororoZeroShotFactory(PororoFactoryBase):
9
10     def __init__(self, task: str, lang: str, model: Optional[str]):
11         super().__init__(task, lang, model)
12
13
14 if "brainbert" in self.config.n_model:
15     from pororo.models.brainbert import BrainRobertaModel
16
17     model = BrainRobertaModel.load_model(
18         f"bert/{self.config.n_model}",
19         self.config.lang,
20     ).eval().to(device)
21     return PororoBertZeroShot(model, self.config)
22
23
24 class PororoBertZeroShot(PororoBiencoderBase): - 분류 수행
```

- 인자로 인스턴스 초기화

- brainbert model을 불러 zero-shot 수행

```
86 def predict(
87     self,
88     sent: str,
89     labels: List[str],
90     **kwargs,
91 ) -> Dict[str, float]:
92     cand = [
93         self._template[self.config.lang].format(label=label)
94         for label in labels
95     ]
96     result = dict()
97     for label, cand in zip(labels, cand):
98         if self.config.lang == "ko":
99             tokens = self._model.encode(
100                 sent,
101                 cand,
102                 add_special_tokens=True,
103                 no_separator=False,
104             )
105
106         else:
107             tokens = self._model.encode(
108                 sent,
109                 cand,
110                 no_separator=False,
111             )
112
113             # throw away "neutral" (dim 1) and take the pr
114             pred = self._model.predict(
115                 "sentence_classification_head",
116                 tokens,
117                 return_logits=True,
118             )[:, [0, 2]]
119             prob = pred.softmax(dim=1)[:, 1].item() * 100
120             result[label] = round(prob, 2)
121
122     return result
```

연구 일자	
연구 제목	
연구 목적	
연구 내용	PORORO Zero-shot Topic Classification Test

테스트1(식품)

-품목: 사조대림 해표 콩기름(식용유) 900ml x 4병

-링크: <http://item.gmarket.co.kr/Item?goodscode=986893111>

-결과

제품 파손시 안내사항

캔 상품 등의 특성상 택배 배송과정에서 불가피하게 일부 살짝 찌그러지는 경우가 있습니다. 이점 구매시 꼭 참고 바랍니다.

② 상품의 단순 찌그러짐 및 스크래치 발생시 반품 및 교환이 어려울 수 있습니다.

③ 심한 파손 발생시, 파손된 제품 사진 촬영바랍니다.

④ 02-599-6966 연락주시면 안내후 파손품처리에 도움 드리도록 하겠습니다.

{ '이름': 9.09, '기능': 17.86, '외형': 39.13, '수상': 14.54, '배송': 85.4, '교환': 18.09 }



{ '이름': 19.26, '기능': 23.08, '외형': 22.34, '수상': 70.87, '배송': 3.11, '교환': 3.7 }



{ '이름': 26.81, '기능': 45.42, '외형': 45.42, '수상': 80.27, '배송': 10.05, '교환': 12.09 }

POINT 1.

콩 100%로 만들어 음식을 고소하게!

콩 100%로 국내에서 직접 만든 맑고 신선한 식용유입니다.

{ '이름': 7.72, '기능': 49.79, '외형': 22.74, '수상': 33.86, '배송': 38.02, '교환': 26.74 }

POINT 3.

다양한 요리에 잘 어울려요

특히 발연점이 높아 바삭함을 유지해야하는 튀김요리에 잘 어울리고, 부침, 볶음 등 다양한 요리에 잘 어울립니다.



{ '이름': 9.79, '기능': 36.16, '외형': 22.6, '수상': 6.12, '배송': 46.1, '교환': 31.59 }

연구 일자	
연구 제목	
연구 목적	
연구 내용	PORORO Zero-shot Topic Classification Test


테스트2(가전제품)

-품목: 쿠쿠홈시스 쿠쿠 CVC-B1020UG

-링크:

<https://search.shopping.naver.com/catalog/29973083619?adId=nad-a001-02-000000165679789&channel=naver.search.pc.npla&query=%EC%B2%AD%EC%86%8C%EA%B8%B0&NaPm=ct%3DI7d5bj4%7Cci%3D0zG00014ceHxZLRyS12U%7Ctr%3Dpla%7Chk%3D16406cb00fca573590fd2392727207c0ac71686&cid=0zG00014ceHxZLRyS12U>

결과




CVC-B1020UG
NEW 2022

강력함에 편리함을 더해
일상을 쾌적하게 바꾸다.

쿠쿠 파워클론

{ '이름': 45.87, '기능': 89.11, '외형': 33.63, '수상': 23.75, '배송': 10.87, '교환': 21.96 }




BLDC모터가 만들어 내는
강력한 사이클론과 흡입력

공간의 품격을
높여주는
슬림핏 매트 블랙

버튼 한 번으로
하리 속일 필요 없이
간편한 먼지배출

더욱 가볍고 편리한
인체공학설계


{ '이름': 21.36, '기능': 65.04, '외형': 38.1, '수상': 15.89, '배송': 14.25, '교환': 14.26 }



공간의 품격을
높여주는
슬림핏 매트 그레이

모던하고 세련된 디자인의
매트 그레이 컬러와 슬림한 바디로
어느 공간에서나 잘 어울립니다.


{ '이름': 16.3, '기능': 23.65, '외형': 36.76, '수상': 11.82, '배송': 3.25, '교환': 5.7 }



분리되는 용브러쉬로
영킨 먼지 쉽게 제거!

용브러쉬는 탈착이 가능하여
가볍게 분리하신 후 관리가
가능합니다.

{ '이름': 6.75, '기능': 67.67, '외형': 24.01, '수상': 12.68, '배송': 9.58, '교환': 14.38 }



브러쉬 분실 걱정없는
부착형 거치대

원하는 곳에 부착하여 편하게 보관 및
충전할 수 있습니다.

{ '이름': 13.53, '기능': 57.81, '외형': 21.36, '구성': 60.07, '수상': 16.38, '배송': 9.88, '교환': 3.7 }

연구 일자	
연구 제목	
연구 목적	
연구 내용	PORORO Zero-shot Topic Classification Test

테스트3(의류)

-품목: AS M NSW AUTHRZD PERSONNEL TE

-링크:

https://shopping.interpark.com/product/productInfo.do?prdNo=9529396624&dispNo=016001&bizCd=P01397&NaPm=ct%3Dl7d643vk%7Cci%3Df3e52d1fbbfee4928a831065882d379c2d62a8fa%7Ctr%3Dsisc%7Csn%3D3%7Chk%3D6beb36df2dfee24b55c22d2f4b445321b0ed38ee&utm_medium=affiliate&utm_source=naver&utm_campaign=shop_20211015_navershopping_p01397_cps&utm_content=conversion_47

-결과


- ※ 나이키 코리아 제품이며 100% 정품을 판매합니다.
- ※ 물류 센터에서 상품을 받아 출고하기 때문에 주문량이 많을 경우, 1~3일 정도 배송 지연 될 수 있으니 시간을 여유있게 두시고 신중한 구매 부탁드립니다. 출고는 선결제 순으로 진행되며 여러 오픈 마켓에서 동시 판매중이므로 주문량이 많은 상품은 출고시간 이후 품절 안내를 받으실 수 있습니다.

{ '이름': 13.82, '기능': 15.41, '외형': 32.94, '구성': 36.84, '수상': 7.81, '배송': 68.21, '교환': 6.51 }

AS M NSW AUTHRZD PERSONNEL TE

품번 DM6428-010
색상 블랙

{ '이름': 32.13, '기능': 29.78, '외형': 68.51, '구성': 57.37, '수상': 42.59, '배송': 35.26, '교환': 51.66 }



상품공지 조명과 카메라의 위치, 각도에 따라 색상 차이가 있을 수 있습니다.

의류는 재질 및 시즌 외 여러가지 이유로 사이즈 차이가 있을 수 있으므로
유선상으로는 자세한 사이즈 추천이 어렵습니다.
가까운 매장에 문의하시거나 착용 및 실물 확인 후 신중한 구매 하시는 것을 권장드립니다.

{ '이름': 4.84, '기능': 25.84, '외형': 31.73, '구성': 43.87, '수상': 11.05, '배송': 19.54, '교환': 9.13, '주의사항': 98.88 }



✓ 구조적이고 독특한 느낌을 선사하는 무게감 있는 면 소재로 제작

{ '이름': 15.0, '기능': 27.1, '외형': 75.68, '구성': 94.72, '수상': 34.9, '배송': 24.32, '교환': 32.61, '주의사항': 64.36 }

연구 일자	
연구 제목	
연구 목적	
연구 내용	PORORO Zero-shot Topic Classification Test

PORORO Zero_shot Topic Classification Test 결론

장점

- 문장 분류를 어느 정도 잘 수행함.
- 기존의 있는 언어 모델로 분류가 가능하여, 추가적인 학습이 필요하지 않음.

단점

- 문장 분류의 성능이 완벽하지 않고, 변수 및 이상값이 많음.
- 기존 언어를 기반으로 분류를 실행하여, 중요도에 따른 분류 등 새로운 규칙을 부여할 수 없음.
- 추가 학습으로 모델 수정이 어려움.
- 분류 기준이 되는 키워드 선정이 주관적임.
- 프로그램이 무거워 한 번 분류를 실행할 때 시간이 꽤 걸림.

결론

- 분류 기능을 잘 지원하고 추가적인 학습이 없어 간편하게 사용할 수 있지만, 세부 변수에 따라 좋지 못한 결과를 내는 등 안정적으로 결과를 내지 못함.
- 모델 수정이 어려워 전이 학습 및 Fine Tuning도 불가능하다. 따라서 PORORO가 지정하지 않은 상황에 유연하게 적용하기 어렵고, 모델 개선 또한 어려워 보임.
- 해당 과제의 문장 분류에 사용할 수는 있음.
- 사용 시 안정적인 기능을 보이기는 어려울 것으로 보임.
- 모델 사용 및 적용의 유연성을 높이기 위해 PORORO에서 사용한 SKT의 KoBERT를 검토 예정.

연구 일자	
연구 제목	
연구 목적	
연구 내용	KoBERT

KoBERT – Korean BERT (Bidirectional Encoder Representations from Transformers)

- 구글 BERT의 한국어 성능 한계를 극복하기 위해 개발
- 위키피디아 등에서 수집한 수백만 개의 한국어 문장으로 이루어진 대규모 말뭉치 학습
- 데이터 기반 토큰화 기법을 적용하여 한국어의 불규칙한 언어 변화의 특성 반영
- 성능 향상
- 파이토치, 텐서플로우 등 다양한 딥러닝 API 지원
- Fine-tuning 가능
- Apache-2.0 라이선스

Architecture

```
predefined_args = {
    'attention_cell': 'multi_head',
    'num_layers': 12,
    'units': 768,
    'hidden_size': 3072,
    'max_length': 512,
    'num_heads': 12,
    'scaled': True,
    'dropout': 0.1,
    'use_residual': True,
    'embed_size': 768,
    'embed_dropout': 0.1,
    'token_type_vocab_size': 2,
    'word_embed': None,
}
```

기술 사이트

sk telecom web site – sktelecom.github.io/project/kobert/
 github -github.com/SKTBrain/KoBERT

연구 일자	
연구 제목	
연구 목적	
연구 내용	KoBERT 실험 설계

1. Fine Tuning 데이터 준비

현재 분류가 완료된 쇼핑 데이터가 존재하지 않음 – 대체 데이터 준비

대체 데이터: KLUE(Korean Language Understanding Evaluation)

기술 사이트

github.com/KLUE-benchmark/KLUE

huggingface.co/datasets/klue

데이터 예시

Dataset Preview				
Subset		Split		
ynat		train		
guid (string)	title (string)	label (class label)	url (string)	date (string)
"ynat-v1_train_00000"	"유튜브 내달 2월까지 크리에이터 지원 공간 운영"	3 (생활문화)	"https://news.naver.com/main/read.nhn?mode=LS2D&mid=shm&sid1=105&sid2=227&oid=001&aid=0000000947"	"2016.06.30. 오전 10:36"
"ynat-v1_train_00001"	"메비미날 뒀다가 돌려줘..남부지방 얼은 활사"	3 (생활문화)	"https://news.naver.com/main/read.nhn?mode=LS2D&mid=shm&sid1=103&sid2=248&oid=001&aid=0000384783"	"2016.05.06. 오전 5:25"
"ynat-v1_train_00002"	"내년부터 국가RD 평가 때 논문건수는 반영 않는다"	2 (사회)	"https://news.naver.com/main/read.nhn?mode=LS2D&mid=shm&sid1=105&sid2=228&oid=001&aid=0000254806"	"2016.03.16. 오후 12:00"
"ynat-v1_train_00003"	"김영자 신임 과총 회장 원로와 젊은 과학자 지혜 모듬 것"	2 (사회)	"https://news.naver.com/main/read.nhn?mode=LS2D&mid=shm&sid1=105&sid2=228&oid=001&aid=0000070646"	"2017.02.28. 오전 9:54"
"ynat-v1_train_00004"	"최혜민간 작가 김동서 양심교백 등 새 소설집 2권 출간"	3 (생활문화)	"https://news.naver.com/main/read.nhn?mode=LS2D&mid=shm&sid1=103&sid2=243&oid=001&aid=0000999529"	"2019.04.03. 오전 7:05"
"ynat-v1_train_00005"	"마워서 생방송 하세요..백선범 전용 요금제 잇따라"	0 (IT과학)	"https://news.naver.com/main/read.nhn?mode=LS2D&mid=shm&sid1=105&sid2=226&oid=001&aid=0000547667"	"2016.07.18. 오전 9:46"
"ynat-v1_train_00006"	"월드컵 미국전사 16강 전초기지 레오강 입성종합"	5 (스포츠)	"https://sports.news.naver.com/news.nhn?oid=001&aid=0010126131"	"2018.06.04 00:15"

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

2. 데이터 재분류

0: IT과학 / 경제 / 사회 / 생활문화

1: 세계 / 스포츠 / 정치

3. KoBERT 모델 Fine Tuning 학습 및 평가

연구 일자	
연구 제목	
연구 목적	
연구 내용	KoBERT 코드

- Data 불러오기

```
dataset = load_dataset("klue", 'ynat')

train_klue = pd.DataFrame({'title': dataset['train']['title'],
                           'label': dataset['train']['label']})

test_klue = pd.DataFrame({'title': dataset['validation']['title'],
                           'label': dataset['validation']['label']})

device = torch.device("cuda:0")
bertmodel, vocab = get_pytorch_kobert_model()
```

- Data 재분류 및 전처리

```
train_data = []
for q, label in zip(train_klue[:5000]['title'], train_klue[:5000]['label']):
    data = []
    data.append(q)
    # 0:IT과학 1:경제 2:사회 3:생활문화
    if label == 0 or label == 1 or label == 2 or label == 3:
        data.append(str(0))
    # 4:세계 5:스포츠 6:정치
    elif label == 4 or label == 5 or label == 6:
        data.append(str(1))
    #data.append(str(label))
    train_data.append(data)

test_data = []
for q, label in zip(test_klue[:]['title'], test_klue[:]['label']):
    data = []
    data.append(q)
    # 0:IT과학 1:경제 2:사회 3:생활문화
    if label == 0 or label == 1 or label == 2 or label == 3:
        data.append(str(0))
    # 4:세계 5:스포츠 6:정치
    elif label == 4 or label == 5 or label == 6:
        data.append(str(1))
    #data.append(str(label))
    test_data.append(data)
```

- 하이퍼파라미터 세팅

```
# Setting parameters
max_len = 64
batch_size = 64
warmup_ratio = 0.1
num_epochs = 20
max_grad_norm = 1
log_interval = 200
learning_rate = 5e-5
```

- 토큰화

```
#토큰화
tokenizer = get_tokenizer()
tok = nlp.data.BERTSPTokenizer(tokenizer, vocab, lower=False)

data_train = BERTDataset(train_data, 0, 1, tok, max_len, True, False)
data_test = BERTDataset(test_data, 0, 1, tok, max_len, True, False)

train_dataloader = torch.utils.data.DataLoader(data_train, batch_size=batch_size, num_workers=5)
test_dataloader = torch.utils.data.DataLoader(data_test, batch_size=batch_size, num_workers=5)

# BERT 모델 불러오기
model = BERTClassifier(bertmodel, dr_rate=0.5).to(device)
```

- 기타 설정

```
# BERT 모델 불러오기
model = BERTClassifier(bertmodel, dr_rate=0.5).to(device)

# optimizer와 schedule 설정
no_decay = ['bias', 'LayerNorm.weight']
optimizer_grouped_parameters = [
    {'params': [p for n, p in model.named_parameters()
                  if not any(nd in n for nd in no_decay)], 'weight_decay': 0.01},
    {'params': [p for n, p in model.named_parameters()
                  if any(nd in n for nd in no_decay)], 'weight_decay': 0.0}
]

optimizer = AdamW(optimizer_grouped_parameters, lr=learning_rate)
loss_fn = nn.CrossEntropyLoss()

t_total = len(train_dataloader) * num_epochs
warmup_step = int(t_total * warmup_ratio)

scheduler = get_cosine_schedule_with_warmup(optimizer,
                                              num_warmup_steps=warmup_step,
                                              num_training_steps=t_total)
```

- calc_accuracy(): 정확도 측정을 위한 함수 정의

- 학습 및 평가

```
for e in range(num_epochs):
    train_acc = 0.0
    test_acc = 0.0
    model.train()
    for batch_id, (token_ids, valid_length, segment_ids, label) in enumerate(tqdm_notebook(train_dataloader)):
        optimizer.zero_grad()
        token_ids = token_ids.long().to(device)
        segment_ids = segment_ids.long().to(device)
        valid_length = valid_length
        label = label.long().to(device)
        out = model(token_ids, valid_length, segment_ids)
        loss = loss_fn(out, label)
        loss.backward()
        torch.nn.utils.clip_grad_norm_(model.parameters(), max_grad_norm)
        optimizer.step()
        scheduler.step() # Update learning rate schedule
        train_acc += calc_accuracy(out, label)
        if batch_id % log_interval == 0:
            print("epoch {} batch id {} loss {} train acc {}".format(e + 1, batch_id + 1,
                                                                      loss.data.cpu().numpy(),
                                                                      train_acc / (batch_id + 1)))
    print("epoch {} train acc {}".format(e + 1, train_acc / (batch_id + 1)))

    model.eval()
    for batch_id, (token_ids, valid_length, segment_ids, label) in enumerate(tqdm_notebook(test_dataloader)):
        token_ids = token_ids.long().to(device)
        segment_ids = segment_ids.long().to(device)
        valid_length = valid_length
        label = label.long().to(device)
        out = model(token_ids, valid_length, segment_ids)
        test_acc += calc_accuracy(out, label)
    print("epoch {} test acc {}".format(e + 1, test_acc / (batch_id + 1)))

# 토큰화
tokenizer = get_tokenizer()
tok = nlp.data.BERTSPTokenizer(tokenizer, vocab, lower=False)
```

- predict(): 토큰화

```
#결과 출력받기! 0 입력시 종료
end = 1
while end == 1:
    sentence = input("하고싶은 말을 입력해주세요 : ")
    if sentence == 0:
        break
    predict(sentence)
    print("\n")
```

연구 일자	
연구 제목	
연구 목적	
연구 내용	KoBERT Test 결과

1. Train data set 50000개

```
epoch 20 train acc 0.9998468137254902
0%|          | 0/143 [00:00<?, ?it/s]
epoch 20 test acc 0.9524694055944056
```

2. Train data set 10000개

```
epoch 20 train acc 0.9997014331210191
0%|          | 0/143 [00:00<?, ?it/s]
epoch 20 test acc 0.9506118881118881
```

3. Train data set 5000개

```
epoch 20 train acc 0.9998022151898734
0%|          | 0/143 [00:00<?, ?it/s]
epoch 20 test acc 0.9390987302171513
```

4. Train data set 1000개

```
epoch 20 train acc 1.0
0%|          | 0/143 [00:00<?, ?it/s]
epoch 20 test acc 0.9409562476996688
```

5. Train data set 500개

```
epoch 20 train acc 1.0
0%|          | 0/143 [00:00<?, ?it/s]
epoch 20 test acc 0.9291958041958042
```

- 결과 확인 예시

하고싶은 말을 입력해주세요 : 우주기원 불 '코리아남' 찾아라...광속 절반 가속' 뉴런급을 한창
세계 / 스포츠 / 정치 중 하나입니다.

하고싶은 말을 입력해주세요 : 위대한 열악사유 불안전 화학공무원...불안' 채용취소' 적법"
뒤늦게 결격사유 발견된 퇴직공무원...법원 "채용취소 적법"
IT과학 / 경제 / 사회 / 생활문화 중 하나입니다.

하고싶은 말을 입력해주세요 : 펜스 앞 미국 부통령 "평창에서 김여정·김영남 일부러 피했다"
펜스 前 미국 부통령 "평창에서 김여정·김영남 일부러 피했다"
세계 / 스포츠 / 정치 중 하나입니다.

하고싶은 말을 입력해주세요 : [카타르 NOW] 카타르, '개최국 무패' 도전...영달아 한국 '4강 신화'도 재조명
[카타르 NOW] 카타르, '개최국 무패' 도전...영달아 한국 '4강 신화'도 재조명
세계 / 스포츠 / 정치 중 하나입니다.

하고싶은 말을 입력해주세요 : 전기차 작년보다 73% 늘었다...국내 등록 친환경차 100만대 돌파
전기차 작년보다 73% 늘었다...국내 등록 친환경차 150만대 돌파
IT과학 / 경제 / 사회 / 생활문화 중 하나입니다.

하고싶은 말을 입력해주세요 : [7시 날씨] 내일 흐리다 자자 맑음...일교차 주의
[7시 날씨] 내일 흐리다 자자 맑음...일교차 주의
IT과학 / 경제 / 사회 / 생활문화 중 하나입니다.

하고싶은 말을 입력해주세요 : '이재명 사법리스크' 현실화에 뿔 '당혹...'후 리더십' 최대 위기
'이재명 사법리스크' 현실화에 뿔 '당혹...'후 리더십' 최대 위기
세계 / 스포츠 / 정치 중 하나입니다.

연구 일자	
연구 제목	
연구 목적	
연구 내용	KoBERT Test 결과

1. KoBERT Test 결과

Train data set	50000	10000	5000	1000	500
Accuracy	0.95	0.95	0.93	0.94	0.92

- 전체적으로 높은 정확도를 보이며, 해당 이진 분류에서 좋은 성능을 보임
- 학습 데이터 수가 높을수록 좋은 정확도를 기록하기는 하나, 적은 데이터 수에서도 비슷하게 좋은 정확도 기록.

2. KoBERT Test 결과 해석

1) 성능 및 안정성

- Test accuracy가 약 90%로 좋은 성능을 보여줌.
- 임의로 넣은 문장에서도 정확한 결과를 보여줌.

2) 유연성

- 코드의 수정이 용이하여 원하는 데이터 셋에 맞게 변경이 용이함.
- Fine Tuning을 통해 특정 데이터 셋에 맞는 모델 학습이 가능함. 학습 데이터의 수가 늘어날수록 좋은 결과를 보여줌.

3) 속도

- 모델 학습 이후, 분류 실행 시 딜레이가 전혀 없음

3. 개선 필요 사항

- 모델 저장 및 불러오기 기능 필요
- Klue 데이터에 대한 실험 결과만 있음. 쇼핑 데이터에도 잘 적용될지 실험 및 확인 필요
- 완전하고 문법에 어긋나지 않은 문장에 대해서만 좋은 결과를 얻음. 문법에 어긋나거나, 여러 문장, 혹은 단어에 대해서도 실험 및 확인 필요

연구 일자	
연구 제목	
연구 목적	
연구 내용	KoBERT 분류 프로그램 기능 추가

KoBERT 저장 및 불러오기 기능 추가

- 모델 학습 및 평가 종료 후 다음 코드 추가

```
# 학습 모델 저장
torch.save(model, 'KoBERT_klue5000_model.pt')
torch.save(model.state_dict(), 'KoBERT_klue5000_model_state_dict.pt') # 모델 객체의 state_dict 저장
torch.save({
    'model': model.state_dict(),
    'optimizer': optimizer.state_dict()
}, 'KoBERT_klue5000_all.tar') # 여러 가지 값 저장, 학습 중 진행 상황 저장을 위해 epoch, loss 값 등 일반 scalar 값 저장 가능
```

- 저장 모델 사용 및 분류 실행 프로그램

* 파라미터 설정

```
#GPU 사용
device = torch.device("cuda:0")

#BERT 모델, Vocabulary 불러오기 함수
bertmodel, vocab = get_pytorch_kobert_model()

# 토큰화
tokenizer = get_tokenizer()
tok = nlp.data.BERTSPTokenizer(tokenizer, vocab, lower=False)

# Setting parameters
max_len = 64
batch_size = 32
```

* 모델 불러오기

```
## 학습 모델 로드
model = torch.load('KoBERT_klue5000_model.pt') # 전체 모델을 통째로 불러옴, 클래스 선언 필수
model.load_state_dict(torch.load('KoBERT_klue5000_model_state_dict.pt')) # state_dict를 불러
```

* predict 함수 – 분류 실행

```
def predict(predict_sentence):
    data = [predict_sentence, '0']
    dataset_another = [data]

    another_test = BERTDataset(dataset_another, 0, 1, tok, max_len, True, False)
    test_dataloader = torch.utils.data.DataLoader(another_test, batch_size=batch_size, num_workers=5)

    model.eval()

    for batch_id, (token_ids, valid_length, segment_ids, label) in enumerate(test_dataloader):
        token_ids = token_ids.long().to(device)
        segment_ids = segment_ids.long().to(device)

        valid_length = valid_length
        label = label.long().to(device)

        out = model(token_ids, valid_length, segment_ids)

        test_eval = []
        for i in out:
            logits = i
            logits = logits.detach().cpu().numpy()

            if np.argmax(logits) == 0:
                test_eval.append("IT관련 / 경제 / 사회 / 생활문화 중 하나입니다.")
            elif np.argmax(logits) == 1:
                test_eval.append("세계 / 스포츠 / 정치 중 하나입니다.")

    print(test_eval[0])
```

연구 일자	
연구 제목	
연구 목적	
연구 내용	쇼핑 설명 분류를 위한 데이터 준비

실제 쇼핑 설명 분류를 위한 데이터 준비 방식

- 1. 예시 물품에 대해 크롤링 및 OCR을 사용하여 쇼핑 전체 설명 수집
- 2. 설명의 각 문장을 나눔
- 3. 설명의 필요 여부 체크

최종 데이터

- 데이터 형태:

label : text

label - 0: 필요하지 않음, 1: 필요함

text - 쇼핑 설명 문장

- 데이터 양:

총 8625개

train data: 80%, test data: 20%

* 이전 실험에서 500개의 데이터를 사용했을 때 약 90%의 정확도를 보인 것으로 보아, 약 8600개의 데이터를 사용한다면 좋은 결과를 기대할 수 있음

행 Label	행 text
1 1	663 다용도 엑스프라 프라이팬 663 TEFLON XTRA COATING FRYPAN 주황의 풍자 독일 663 브랜드 가 당신의 주방의 풍격을 높이 준
2 0	POWER IMPACT BOTTOM + INDUCTION
3 0	1
4 0	R09 663 SCHTOPF
5 1	사용하는 시점의 건강까지 생각한 독일의 기술력, 오랜 시간 사용해 도 변질되지 않은 강한 내구성과 고급스러운 디자인으로 40여 년간
6 0	100%
7 0	663 KUCHENTOPF Designed in Germany
8 0	T
9 0	주방의 풍자 독일 663 브랜드 독일 663 브랜드는?
10 0	BE
11 0	() MOLLONICWESSON
12 0	663
13 0	663 KOREA
14 0	scratch - resistant function !
15 0	Teflon " on test stick
16 0	Long Life Easy Clean Convenien
17 0	선택 62
18 0	663 다용도 엑스프라 프라이팬 28cm
19 0	규격 : 28cm 재질 : 스테인리스 스틸
20 0	선택 62
21 0	663 다용도 엑스프라 프라이팬 26cm
22 0	규격 : 26cm 재질 : 스테인리스 스틸
23 0	선택 62
24 0	663 다용도 엑스프라 프라이팬 28cm
25 0	규격 : 28cm 재질 : 스테인리스 스틸
26 0	선택 64
27 0	663 다용도 엑스프라 프라이팬 30cm
28 0	규격 : 30cm 재질 : 스테인리스 스틸
29 0	선택 65
30 0	663 다용도 엑스프라 프라이팬 2P 세트 28cm + 28cm

데이터 형태 예시

연구 일자	
연구 제목	
연구 목적	
연구 내용	쇼핑 설명 분류 결과

1. 실험 방법

- 1) 수집 및 라벨링 완료 데이터 준비.
- 2) 준비된 학습 데이터를 사용하여 KoBert 모델 Fine Tuning 진행.
- 3) 테스트 데이터를 사용하여 테스트 진행.

2. 쇼핑 설명 분류 결과

Data	Accuracy
Train Data	0.999
Test Data	0.907

```
epoch 20 train acc 0.9994212962962963
0%|          | 0/27 [00:00<?, ?it/s]
epoch 20 test acc 0.907265103217972
```

분류 결과 표 및 캡처 그림

3. 실험 결과 해석

- 1) 성능 및 안정성
 - 테스트 데이터에 대한 예측 정확도를 약 90%를 기록함.
 - 임의의 문장에 대해서도 좋은 성능을 보임.
- 2) 속도
 - 임의의 문장(상품 설명)을 모델에 입력한 경우 훌륭한 속도를 보임.
 - 실제 측정 필요.
- 3) 결론
 - 실험에서는 쇼핑 설명의 중요도 분류를 성능 및 속도 측면에서 좋은 결과를 보여줌.
 - 분류 결과 직접 검토를 통해 모델 성능에 대한 크로스 체크 필요.