(딥러닝 응용)

뉴스 요약 및 카테고리 분류 모델

이승은 정현정 박지영 안성하

(CONTENTS)

〈 프로젝트 주제 / 배경 〉

" 뉴스 요약 및 카테고리 분류 모델 "

현대 사회 뉴스와 정보의 중요성이 높아지고 있지만
대량의 뉴스 기사를 읽는 것은 많은 시간과 노력 필요.
아에 뉴스를 요약하여 필요한 정보를 효과적으로 제공할 수 있다고 판단

〈 사용모델 〉

Transformer 모델
양방향 Seq2Seq 구조를 가지고 Attention을
사용해 텍스트의 문맥 파악

Self-Attention
스스로의 정보만을 사용하여 데이터별 중요도를 판별

Positional Encoding
단어의 순서 정보를 고려하지 않기 때문에 입력에 위치 정보를 추가

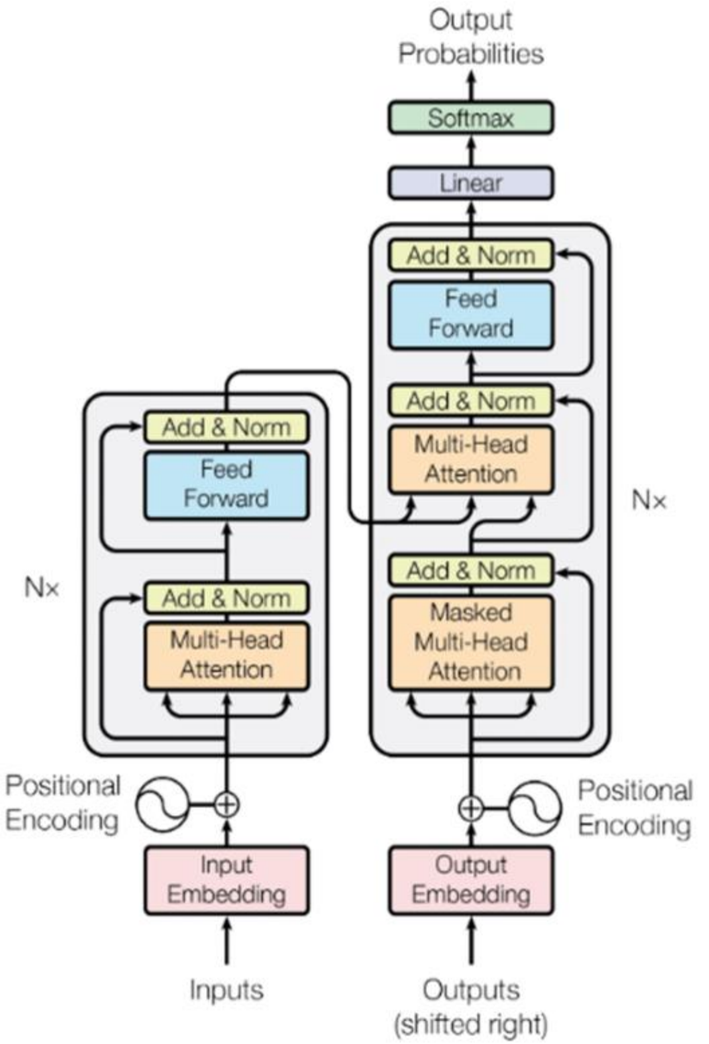


Figure 1: The Transformer - model architecture.

〈 사용모델 〉

Transformer 모델
양방향 Seq2Seq 구조를 가지고 Attention을
사용해 텍스트의 문맥 파악

Self-Attention
스스로의 정보만을 사용하여 데이터별 중요도를 판별

Positional Encoding
단어의 순서 정보를 고려하지 않기 때문에 입력에 위치 정보를 추가

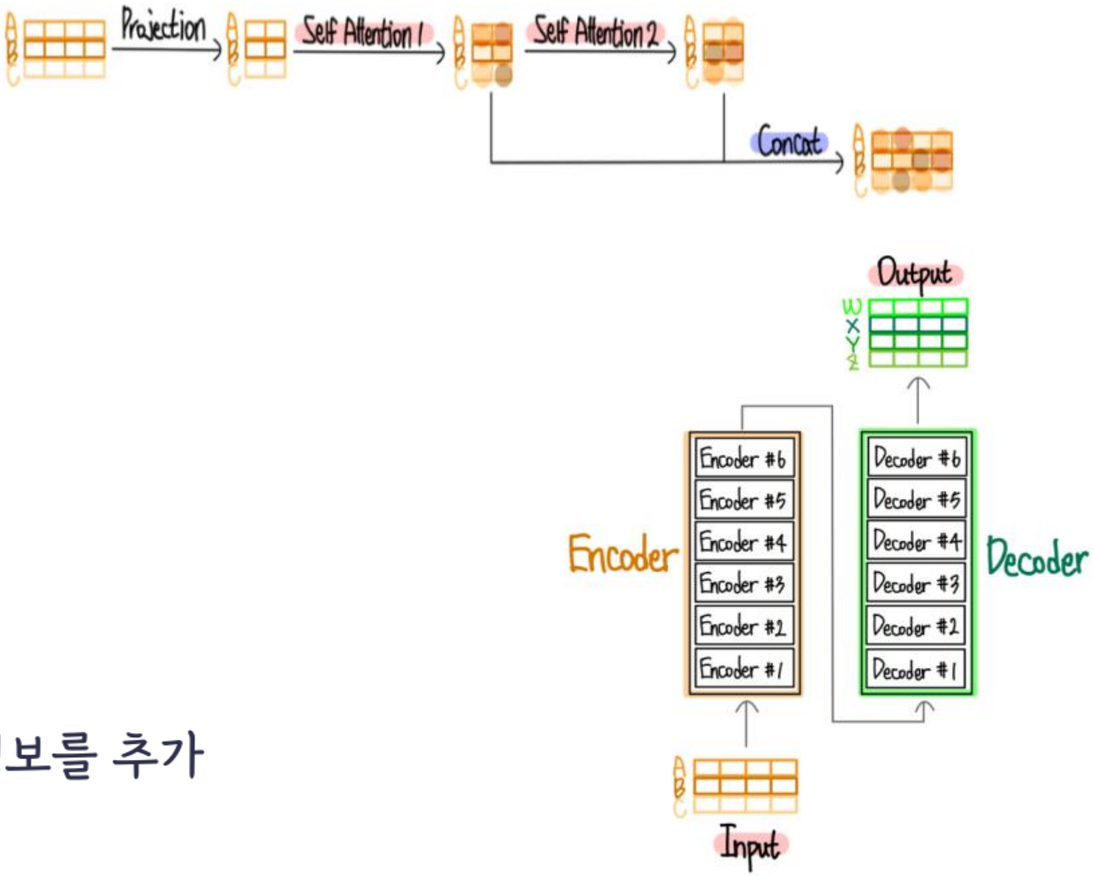


그림8. Transformer Multi Head Self Attention 구성

〈 사용모델 〉

Transformer 모델
양방향 Seq2Seq 구조를 가지고 Attention을
사용해 텍스트의 문맥 파악

Self-Attention
스스로의 정보만을 사용하여 데이터별 중요도를 판별

Positional Encoding
단어의 순서 정보를 고려하지 않기 때문에 입력에 위치 정보를 추가

Encoder

Decoder

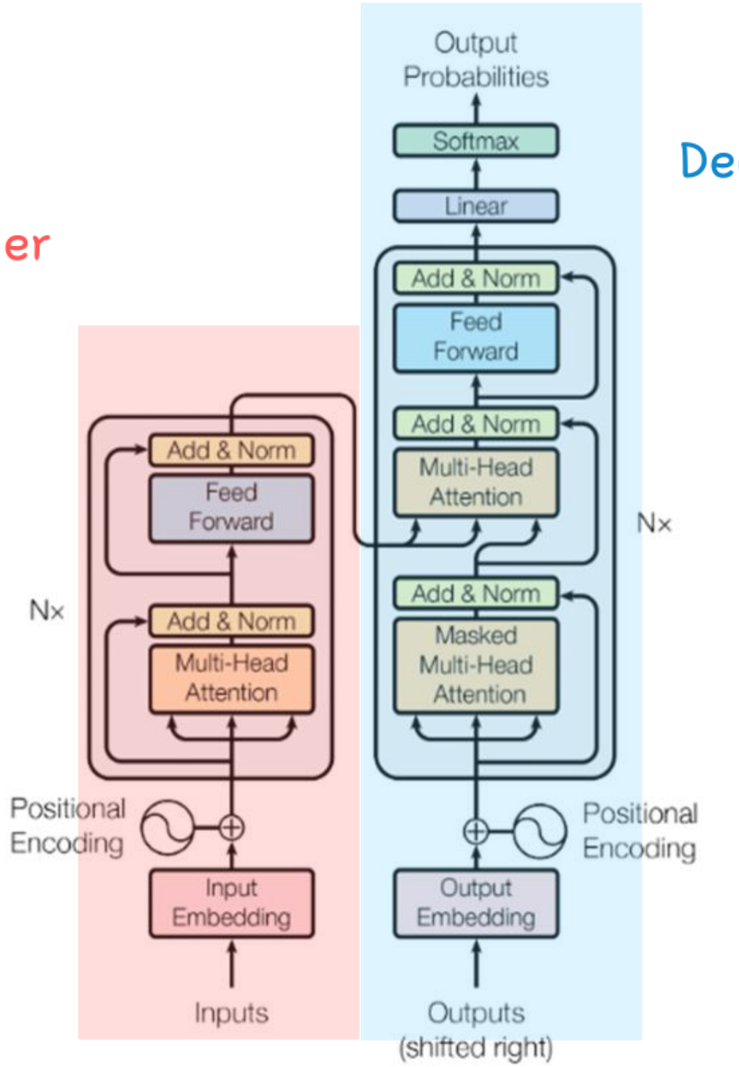


Figure 1: The Transformer - model architecture.

Transformer 모델 장점

1) 한 덩어리의 입력 데이터 받기

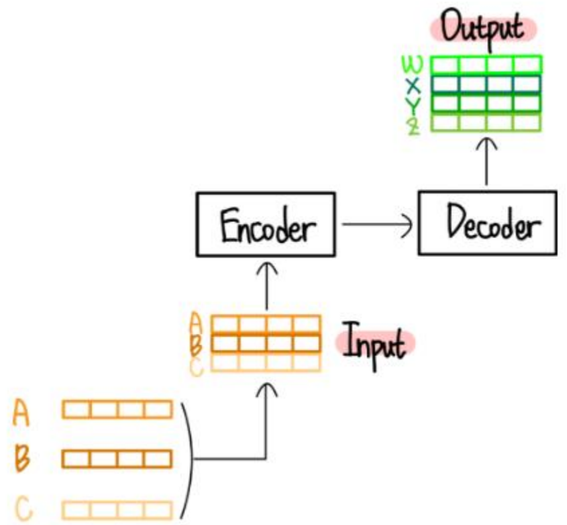


그림2. Transformer의 입력 데이터

계산 속도 향상 및 정보 소실 막음

2) Self Attention 적용

그림3. Self Attention 연산 전후

스스로의 데이터만으로 중요도를 판단
=〉 긴 데이터도 효과적 처리 가능

〈 데이터셋 및 전처리 〉

데이터셋 구성 : 104,053개

AI허브 문서 요약 데이터 100,000 +

네이버 뉴스 크롤링 데이터 4,053

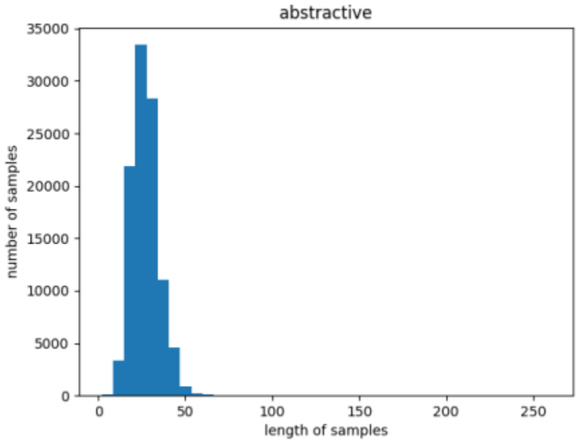
데이터 전처리

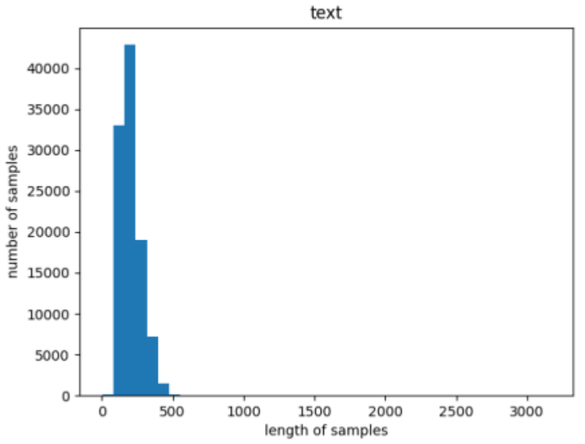
텍스트와 요약 데이터를 불러와

PreTrainedTokenizerFast 토크나이저를 전처리

.joson -〉 .tsv 메모리 사용량 줄임

```
전체 샘플수 : 104053
id              0
text            0
abstractive     0
dtype: int64
결측치 제거 샘플수 : 104053
----------길이------------
텍스트의 최소 길이 : 2
텍스트의 최대 길이 : 3166
텍스트의 평균 길이 : 205.93414894332696
요약의 최소 길이 : 2
요약의 최대 길이 : 260
요약의 평균 길이 : 27.057970457363073
----------전체 비율------------
전체 샘플 중 길이가 1024 이하인 샘플의 비율: 0.9998846741564396
전체 샘플 중 길이가 64 이하인 샘플의 비율: 0.9992119400690033
```





〈 모델 / 모델 훈련 - 파라미터 설정 〉

카테고리 분류 - RoBERTa

```
device = torch.device("cuda")

model = RobertaForSequenceClassification.from_pretrained(
    "klue/roberta-large", num_labels=7).to(device)

epochs = 3
batch_size = 128

optimizer = AdamW(model.parameters(), lr=1e-5)
train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=batch_size, shuffle=False)
test_loader = DataLoader(test_dataset, batch_size=batch_size, shuffle=False)
```

Epoch	Max_length	Batch Size	Val Accuracy	Train Time
3	27	128	GPU 메모리 오버	
3	20	128	0.8820	18m
5	20	64	0.8837	35m
5	23	64	0.8817	40m
3	23	64	0.8851	27m

epochs ,batch_size:

-〉 여러 조합을 돌려보았을 때,

정확도가 가장 높고 효율이 좋은 파라미터로 선정

〈 모델/모델 훈련 - Train 모델 학습 〉

카테고리 분류 - RoBERTa

```
# train
losses = []
accuracies = []

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

  total_loss = 0.0
  correct = 0
  total = 0

  for input_ids_batch, attention_masks_batch, y_batch in tqdm(train_loader):

    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)
  torch.cuda.empty_cache()
```

```
100% 286/286 [
/usr/local/lib/python3.10/dist-packages/transformers/toker
  warnings.warn(
Train Loss: 0.004054039862374403 Accuracy: tensor(0.8299,
100% 286/286 [
Train Loss: 0.003180471893988587 Accuracy: tensor(0.8644,
100% 286/286 [
Train Loss: 0.002730820938118085 Accuracy: tensor(0.8831,
```

에폭(=3)마다 모델 학습, 학습 중 손실 & 정확도 기록

특징

- 1에폭마다 286개의 배치 학습

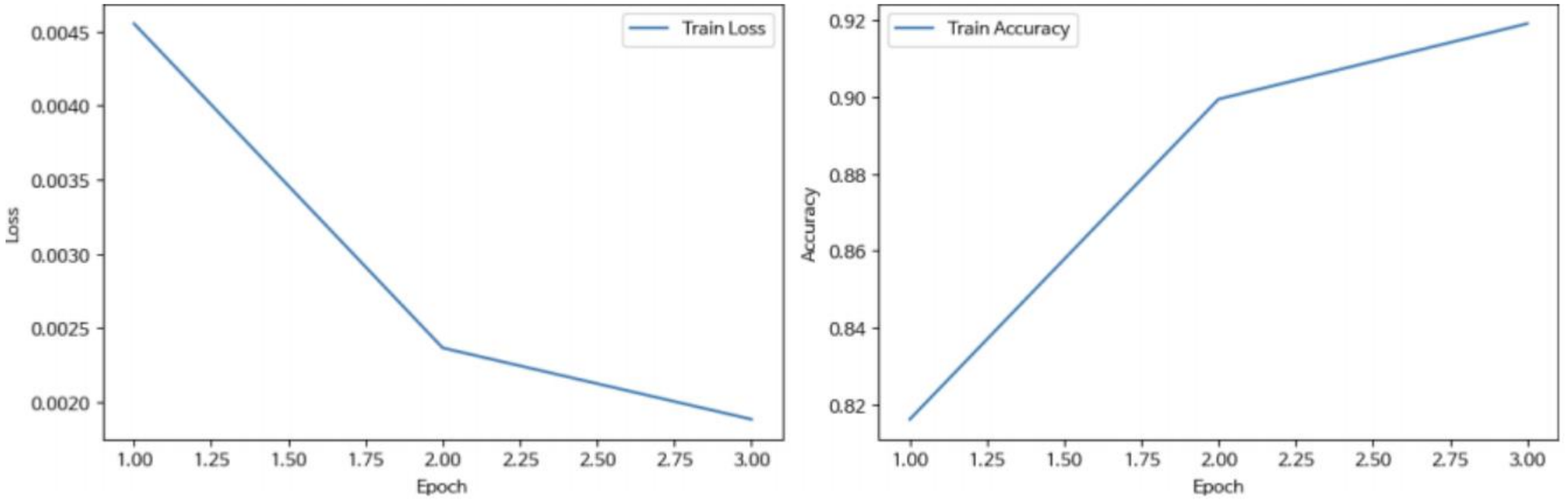
*Train data 36523 / Batch size 128 = 286

- 손실함수 : 교차 엔트로피(두 확률분포 간의 차이)
- 역전파 수행 : 출력값과 실제값 사이의 오차를 역으로 전파

-〉 각 계층의 가중치와 편향을 조정

〈 모델/모델 훈련 - Loss, Accuracy 〉

카테고리 분류 - RoBERTa



손실값 : [0.00455, 0.00236, 0.00188] / 정확도 : [0.8163, 0.8994, 0.9190]

=〉 에폭이 진행될수록 손실값 감소 / 에폭이 진행될수록 정확도 증가

〈 모델 / 모델 훈련 - 파라미터 설정 〉

뉴스 요약 - KoBART

모델 및 토크나이저 설정

BART-ko-mini 모델 및 토크나이저 로딩

PyTorch 기반의 모델을 GPU로 이동

데이터셋 및 데이터로더 설정

배치 크기 64로 설정하고 모델에 공급

옵티마이저 및 손실 함수 설정

AdamW 옵티마이저와 크로스엔트로피 손실함수를 사용해 모델 학습

그래디언트 누적 기법을 사용해 메모리 효율성 높임

KoBERT Architecture

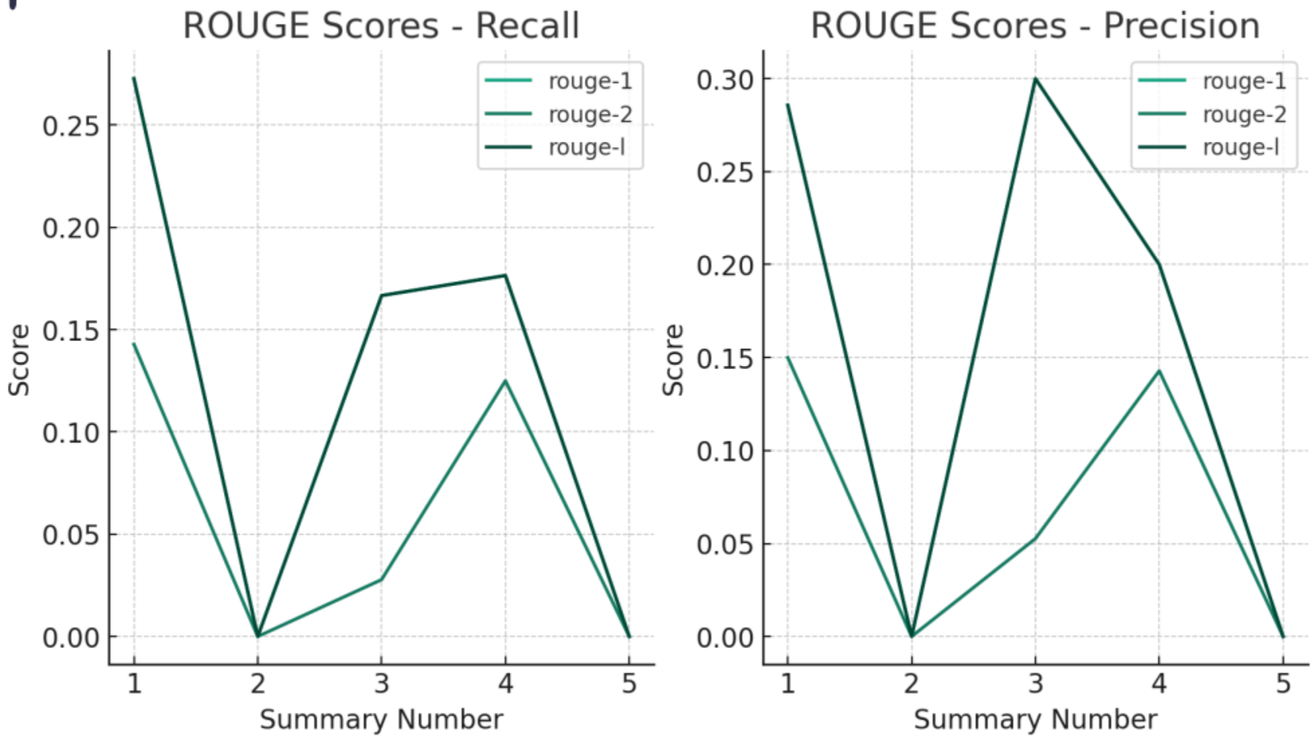
```
predefined_args = {
        'attention_cell': 'multi_head',
        'num_layers': 12,
        'units': 768,
        'hidden_size': 3072,
        'max_length': 512,
        'num_heads': 12,
        'scaled': True,
        'dropout': 0.1,
        'use_residual': True,
        'embed_size': 768,
  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

- 학습셋

데이터	문장	단어
한국어 위키	5M	54M

〈 모델/모델 훈련 - Loss, Accuracy 〉

뉴스 요약 - KoBART



기계가 인식하는 유사도가 0.2 수준으로 높지 않음

=〉 정확도로 판단이 아닌 네이버 뉴스 크롤링 봇으로 비교 예정

〈 개선 전 / 후 〉

BART-ko-mini

〈개선 전〉

1) 종결어미 문제

"신혼부부가 10쌍 중 3쌍 꼴인 것으로 집계됐다다다다다다"

2) 본문이 요약 없이 그대로 출력

〈개선 후〉

요약 데이터 셋 30,000개 추가 및 파인튜닝

"신혼부부가 10쌍 중 3쌍 꼴인 것으로 집계됐다"

〈 결과 〉

```
C:\Users\pjy01\anaconda3\envs\DeeplearningPlatform_2023_fall\python.exe C:\Users\pjy01\DeeplearningPlatform_2023\answer_seq.py
The tokenizer class you load from this checkpoint is not the same type as the class this function is called from. It may result in unexpected tokenization.
The tokenizer class you load from this checkpoint is 'BartTokenizer'.
The class this function is called from is 'PreTrainedTokenizerFast'.
뉴스 본문을 입력하세요: 라온피플(대표 이석중)은 뷰티플렉스와 MOU를 체결하고, 인공지능(AI) 피부분석 및 진단 솔루션 '아이미모(AIMIMO)'를 출시했다고 18일 밝혔다. 아이미모는 한번의 촬영으로 피
---------
요약봇을 입력하세요: 라온피플은 뷰티플렉스와 MOU를 체결하고, 인공지능 피부분석 및 진단 솔루션 '아이미모'를 출시했다고 18일 밝혔다. 양재 엘타워에서 열린 아이미모 론칭 및 MOU행사에는 라온피플
---------
네이버 요약 봇:  라온피플은 뷰티플렉스와 MOU를 체결하고, 인공지능 피부분석 및 진단 솔루션 '아이미모'를 출시했다고 18일 밝혔다. 양재 엘타워에서 열린 아이미모 론칭 및 MOU행사에는 라온피플 이석
우리조 요약 봇:  아토피피플(대표 이석중)은 뷰티플렉스와 MOU를 체결하고, 인공지능(AI) 피부분석 및 진단 솔루션 '아이미모(AIMIMO)'를 출시했다고 18일 밝혔다. 아이미모는 한번의 촬영으로 피부상태
```

Kobart 모델에 Self-Attention Layers를 포함하여
문장 내 각 단어의 관계를 더 잘 파악할 수 있도록 설계

(딥러닝 응용)

감사합니다

이승은 정현정 박지영 안성하