Chapter 8.
Sentence-BERT 및 domain-BERT 살펴보기

발표자: 박채원

22-09-23

들어가기 전

- Sentence-BERT
  - 작동 원리
  - 문장 표현 계산 방법
  - Sentence-transformers 라이브러리
- 지식 증류로 다국어 임베딩 학습
- Domain-BERT (도메인에 특화된 BERT)
  - ClinicalBERT 및 BioBERT
  - ClinicalBERT: 학습 방법 및 재입원 확률 계산
  - BioBERT: 학습 방법 및 개체명 인식 및 질문-응답 태스크로 파인튜닝 하는 방법

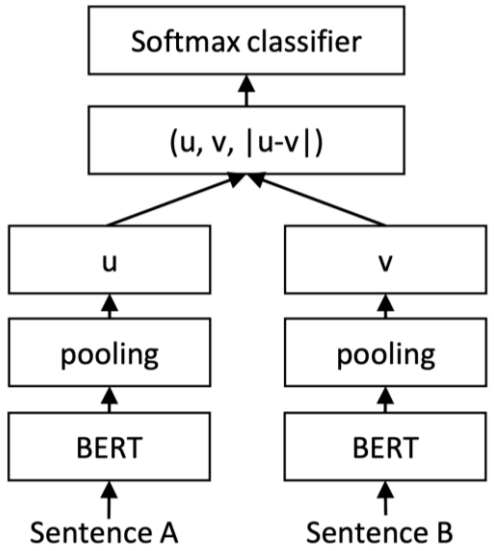
8.1 Sentence-BERT로 문장 표현 배우기

- 유비쿼터스 지식 처리 연구소에서 만들어짐
- 고정 길이의 문장 표현을 얻는 데 사용됨
- Sentence BERT를 사용하는 이유
  - Vanilla bert는 추론하는 데 시간이 오래 걸림
    ex) 문장이 많은 데이터셋에서 서로 유사도가 높은 문장 쌍을 찾으려면 시간이 오래 걸림
- 문장 표현 계산
  - 문장을 토큰화 한 후 사전학습 된 BERT에 입력했을 때 출력되는 **CLS 토큰**의 표현이 문장의 총체적 표현을 가지고 있다고 봄
  - 하지만 특히 모델을 파인튜닝하지 않고 사용할 때는 이 CLS 토큰이 문장 표현이라고 보기 어려워진다.
  - 이 경우 CLS 토큰의 표현 대신 **풀링**을 사용
    - 최대 풀링: 본질적으로 중요한 단어(토큰)의 의미를 가짐
    - 평균 풀링: 본질적으로 모든 단어(토큰)의 의미를 가짐

8.1 Sentence-BERT로 문장 표현 배우기

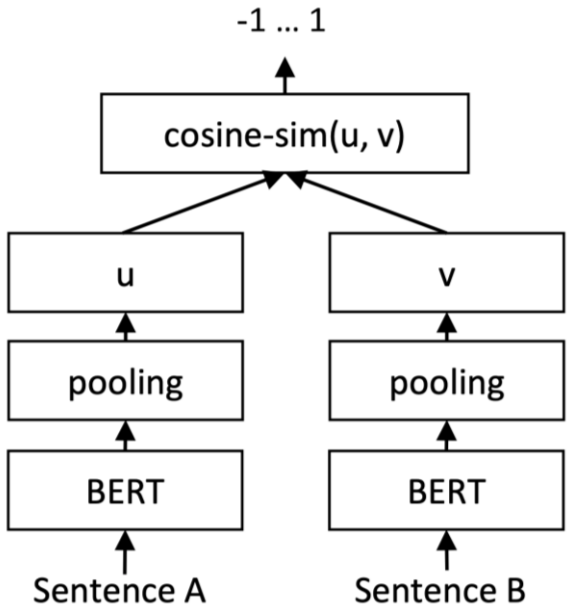
- sentence-BERT 이해하기
  - 사전학습 된 BERT를 택해 문장 표현을 얻도록 파인튜닝함
  - 즉 sentence-BERT는 문장 표현을 계산하기 위해 파인 튜닝된 사전 학습 BERT 모델이다
  - 특별한 점: 파인튜닝 시 **샴 및 트리플렛 네트워크 아키텍처**를 사용하므로 더 빠르게 파인튜닝 가능

- 문장 쌍 분류 태스크를 위한 sentence-BERT
  - 두 문장이 유사한지 유사하지 않은지 분류하는 태스크 (1,0 이진 분류)
  - 동일한 가중치를 공유하는 사전 학습된 BERT 모델을 사용함
  - 각각에 두 문장을 넣고 평균 풀링을 사용해 최종 문장 표현을 얻는다
  - 출력: $softmax(W_t(u, v, |u - v|))$
  - 교차 엔트로피 손실을 최소화하도록 가중치 업데이트



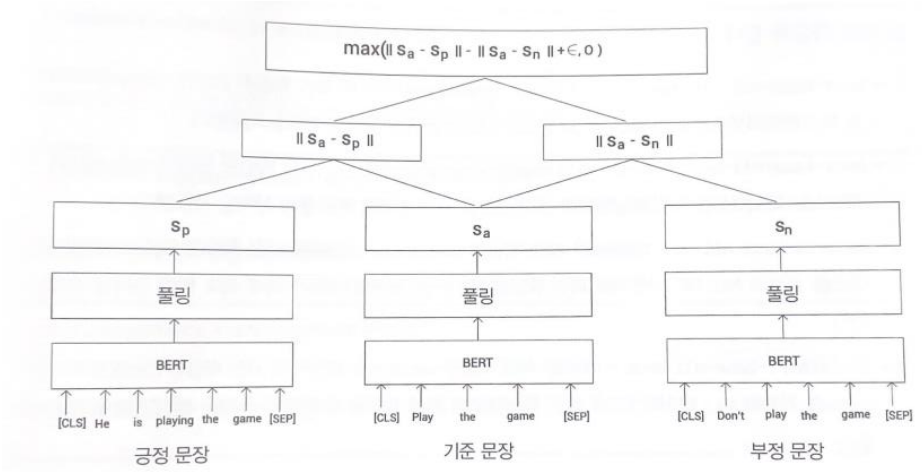
8.1 Sentence-BERT로 문장 표현 배우기

- 문장 쌍 회귀 태스크를 위한 sentence-BERT
  - 문장의 유사도 점수(ex 0.99, 0.00)가 라벨임
  - 사전학습 된 BERT 각각에 문장을 입력해서 각 토큰의 표현을 얻고, 풀링을 적용해 문장의 표현을 얻음
  - 유사도 = $\cos(u, v)$ 문장 표현 간의 유사도를 계산
  - 평균 제곱 손실을 최소화하고 모델의 가중치를 업데이트함



8.1 Sentence-BERT로 문장 표현 배우기

- 트리플렛 네트워크 sentence-BERT
  - 기준 문장, 긍정 문장, 부정 문장 – 문장 3개 -> 트리플렛 네트워크 아키텍처 사용
  - 기준과 긍정은 유사도가 높아야 하며, 기준과 부정은 유사도가 낮아야 한다.
  - 트리플렛 목적함수
    - $\max(\| S\_a - S\_p \| - \| S\_a - S\_n \| + \varepsilon, 0)$
    - $\|\cdot\|$ 는 거리 메트릭(유클리디안 거리)을 나타냄.
    - 이를 최소화하면 긍정 문장과 기준 문장 사이의 유사도가 부정 문장과 기준 문장 사이의 유사도보다 커진다.
    - 즉 기준과 긍정 사이 거리를 좁히고, 기준과 부정 사이 거리를 넓힌다.



8.2 sentence-transformers 라이브러리 탐색

- Pip install –U sentence-transformers 로 라이브러리 설치
- 사용한 BERT 모델, 문장 표현 사용 방법, 파인튜닝 태스크 방법 등에 따라 여러 모델 존재

```
from sentence_transformers import SentenceTransformer

model = SentenceTransformer("bert-base-nli-mean-tokens")
sentence = "paris is a beautiful city"
sentence_representation = model.encode(sentence)

print(sentence_representation.shape)
```

```
(768,)
```

문장 표현 계산

```
import scipy
from sentence_transformers import SentenceTransformer, util

model = SentenceTransformer("bert-base-nli-mean-tokens")

sentence1 = "It was a great day"
sentence2 = "Today was awesome"

sentence1_representation = model.encode(sentence1)
sentence2_representation = model.encode(sentence2)

cosine_sim = util.pytorch_cos_sim(sentence1_representation,sentence2_representation)
cosine_sim
```

```
tensor([[0.9313]])
```

문장 표현 계산 후 유사도 계산

Bert-base-nli-mean-tokens: 사전학습 된 bert를 개체명 인식 태스크로
파인튜닝 하고, 평균 풀링으로 문장 표현을 얻는 모델

8.2 sentence-transformers 라이브러리 탐색

```python
from sentence_transformers import models, SentenceTransformer

word_embedding_model = models.Transformer("albert-base-v2")
pooling_model = models.Pooling(word_embedding_model.get_word_embedding_dimension(),
                               pooling_mode_cls_token=False,
                               pooling_mode_max_tokens=False,
                               pooling_mode_mean_tokens=True)
model = SentenceTransformer(modules=[word_embedding_model, pooling_model])
model.encode("Transformers are awesome")
```

커스텀 모델 로드

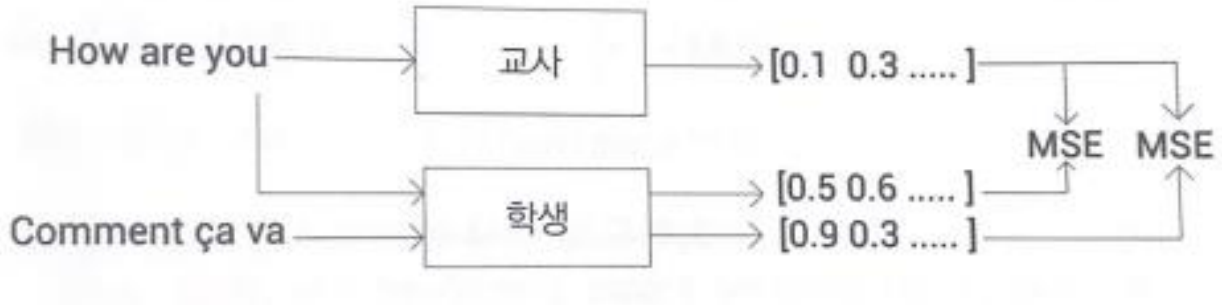
```python
from sentence_transformers import SentenceTransformer, util
import numpy as np

model = SentenceTransformer("bert-base-nli-mean-tokens")

master_dict = ['How to cancel my order?',
               'please let me know about the cancellation policy?',
               'Do you provide refund?',
               'what is the estimated delibery date of the product?',
               'why my order is missing?',
               'how do i report the delivery of the incorrect items?']

inp_question = 'When is my product getting delivered?'
inp_question_representation = model.encode(inp_question, convert_to_tensor=True)
master_dict_representation = model.encode(master_dict, convert_to_tensor=True)

similarity = util.pytorch_cos_sim(inp_question_representation, master_dict_representation)

print("The most similar question in the master dictionary to given input question is:", master_dict[np.argmax(similarity)])
```

```
The most similar question in the master dictionary to given input question is: what is the estimated delibery date of the product?
```

Sentence-BERT로 유사한
문장 찾기

8.3 지식 증류를 이용한 다국어 임베딩 학습

- 지식 증류를 통해 단일 언어 임베딩을 다국어 문장에 적용
- M-BERT, XLM, XLM-R
  - 다른 언어로 된 동일한 문장의 표현이 벡터 공간에서 다른 위치로 매핑됨
  - Sentence-BERT에 대한 지식을 XLM-R과 같은 다국어 모델에 전달하고 다국어 모델이 사전 학습된 sentence-BERT와 동일한 임베딩을 형성하도록 함
  - Sentence-BERT를 교사로 사용, 사전학습 된 XLM-R을 학생 모델로 사용



  - 교사의 단일 언어 임베딩이 생성된 방법과 동일하게 학생이 다국어 임베딩을 생성하도록 할 수 있음

8.3 지식 증류를 이용한 다국어 임베딩 학습

- 다국어 모델 사용
  - Distiluse-base-multilingual-cased: 13개의 언어를 지원하는 모델

```
from sentence_transformers import SentenceTransformer, util
import scipy

model = SentenceTransformer("distiluse-base-multilingual-cased")
eng_sentence = 'thank you very much'
fr_sentence = 'merci beaucoup'

eng_sentence_embedding = model.encode(eng_sentence)
fr_sentence_embedding = model.encode(fr_sentence)


similarity = util.pytorch_cos_sim(eng_sentence_embedding, fr_sentence_embedding)
print("The similarity score is:", similarity)
```

```
The similarity score is: tensor([[0.9840]])
```

8.4 domain-BERT

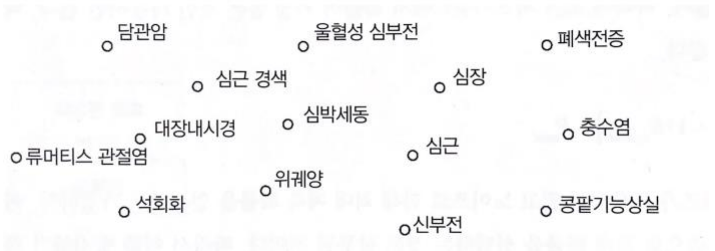
- 특정 도메인 데이터만을 사용하는 것은 BERT가 특정 도메인 임베딩을 학습시키는 데 도움이 된다.
- ClinicalBERT, BioBERT 두가지를 살펴볼 예정

- ClinicalBERT
  - 대규모 임상 말뭉치(MIMIC-III)에서 사전 학습된 임상 domain-BERT 모델
  - 재입원 예측, 체류 기간, 사망 위험 추정, 진단 예측 등과 같은 다양한 다운스트림 태스크에 사용됨
  - MLM과 NSP 태스크를 이용해 사전학습 됨

  - 임상 메모를 입력으로 넣고 CLS토큰의 표현을 가져와 분류기(피드포워드+시그모이드)에 입력해 재입원할 확률 등을 출력할 수 있다. (분류 태스크)
  - 만약 임상 메모의 길이가 BERT의 최대 토큰 길이인 512를 넘는다면 이를 서브시퀀스로 분할 해 개별적으로 예측해 점수 계산
  - $P\left(readmit = 1 \middle| h_{patient}\right) = \frac{P_{max}^{n} + \frac{P_{mean}^{n} * n}{c}}{1 + \frac{n}{c}}$

8.4 domain-BERT

- 임상 단어 유사도 출력
  - 의학 용어 표현 계산 후 t-SNE를 이용해 표현을 도표화
  - 보면 관련된 의학 용어끼리 가깝게 있는 걸 확인할 수 있다
    -> clinicalBERT의 표현이 의학 용어에 대한 콘텍스트 정보를 갖고있다.



- BioBERT
  - 대규모 생물 의학 코퍼스(PubMed, PubMed Central)에서 사전 학습된 생물 의한 domain-BERT
  - 사전학습 전 사전학습된 일반 BERT로 가중치를 초기화하고, 의학 도메인 말뭉치를 사용해 사전학습
  - 워드피스토크나이저 사용, 대소문자 有
  - 개체명 인식 태스크
    - 질병, 약물 등의 클래스 등으로 개체명 인식 필요
    - 출력된 토큰 표현들을 분류기(피드포워드 네트워크+소프트맥스 함수)에 입력하면 개체명 출력

마무리

- Sentence-BERT의 작동 원리 이해
  - 표현을 계산하기 위한 방법 (평균/최대 풀링)
- 트리플렛 네트워크 (긍정, 기준, 부정)
- Sentence-transformers 라이브러리를 사용해 문장 표현 얻기
- 지식 증류를 이용해 다국어 언어 모델에 sentence bert의 임베딩을 학습 시키는 방법
- Domain-BERT 모델
  - ClinicalBERT, BioBERT