NLP embedding 보고서

1. 방향성

[1순위: 한국어에서 특히 그런데, 평범한 질의 문장에도, 서로 다른 문장구조나 형태로 구성되어 있는 경우가 많다. 결론적으로 똑 같은 의미로 받아들여야 하는 문장도 토큰으로 쪼개 임베딩을 하다보면 결국 최종 벡터화 한 부분에서 문장단위로 봤을 때 똑 같은 두문장이 서로 다른 의미의 문장으로 해석될수 있기 때문에 이부분을 신경쓰면서 임베딩 할 수 있는 방법을 찾아야 할것같다.

이 방법을 찾기위해 임베딩 모델이 어떻게 이루어져있는지, 임베딩을 하는방법, 임베딩의 최종목표, 결론적으로 이 임베딩 모델을 어떻게 미세조정 할건지가 관건일것같다. ]

1. 시작

[임베딩 모델이 하는일]

* 단어를 정수화 시키고, 테이블로 만들어 임베딩 벡터를 만든다.

정수화 시킨 단어를 어떤방식으로 벡터로 만들지가 임베딩 벡터의 모델이 하는일이다.<

<https://huidea.tistory.com/168> ->여러가지 임베딩 모델들이 어떻게 학습을 진행하는지, 어떤기준으로 벡터화 하는지>

[임베딩모델별 임베딩을 하는 기준]

Word2vec:단어간 벡터 내적 거리 기준

LRA:문서단어 행렬후 특이값분해Svd적용

Glove:

\*\*Transformers, Bert: 결국 최종적으로는 문장의 context화-> <https://k-min-algorithm.tistory.com/37>(자세한 임베딩순서)

예시) Bert모델을 사용한 챗봇이라고 하자 사용자가 input으로 “나는 고양이를 좋아해”를 입력하면 . 모델에 들어가기전 이 자연어 문장을 토큰화하여 [“나”,”는”,”고양이”,”를”,”좋”,”아”,”해”]로 분리되고, 각각의 토큰들은 Bert모델의 임베딩 공간에 매핑된다, 여기서 벡터형태로 표현되고, 초기에는 무작위 값에서 시작하지만, 학습과정을 통해 조정된다. 그다음 이 임베딩된 토큰들은 tranformer아키텍처 내에서 각 단어간 상호작용을 모델링 하여, 문맥정보를 계산한다. transfomer아키텍처는 여러 layer로 구성되어져 있고, 상위층으로 갈때마다 이 문맥정보가 심화된다. 모델마다 설정된 transformer의 layer값을 다 통과하고 나면, 이 문장에 대한 최종 context된 정보가 나온다.

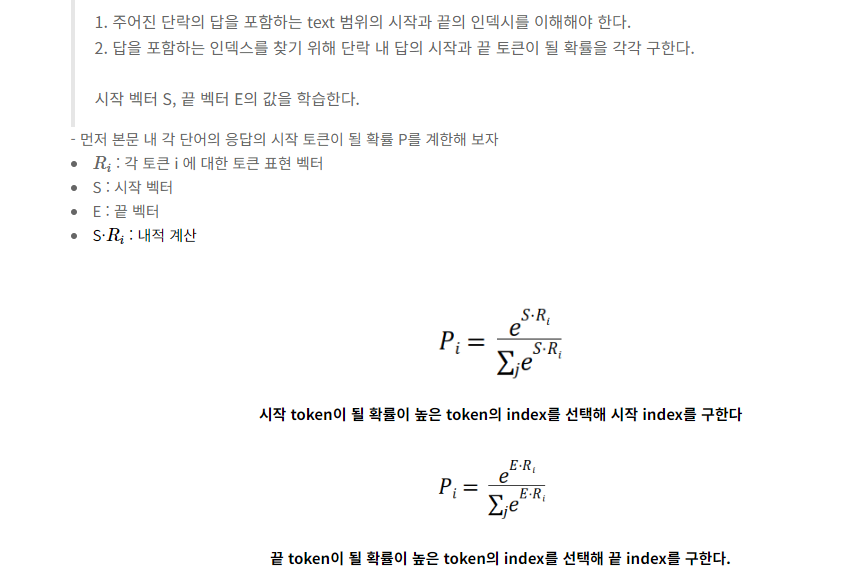
+ Bert 임베딩 과정을 구현해보기

1.rt 임베딩 과정 구현 테스트(bert-base-uncased 모델(pretrained) 사용)

Cls + I + love+ paris + sep + pad+ pad

(상세과정은 코드주석)

1. Bert를 활용해 질의 응답을 할수있게 모델을 파인튜닝하는 과정 구현 테스트
2. 학습과정: 과 본문이 주어지면(input), 본문에서 답을 찾아 답변하는 것
3. 파인튜닝:



다른 모델들도 파인튜닝하기위해 일단 Bert모델이 어떻게 임베딩되고, 이 코드가 뭐고, 어떤모델을 써서 파인튜닝 시킬수있는지 시도해보기위한 목적.

<트장님이 보내주신자료:벤치마킹 기준에 따라 분류된 임베딩 모델들>

+미세조정 Lora, peft && 동일한질문 다른 문장에 대한 데이터셋 학습

E5가 성능이 좋고, 오픈소스여서 우리 상황에 맞게 파인튜닝 할수있다.

[E5성능 테스트] 성능테스트 완료 f1score,accuracy 등 사용 지표

Accelerate: gpu를 효율적으로 쓸수있게 할 수 있는 라이브러리

Peft(fine-tunning):

Koalphaca:

Openai embedding

Google text

[임베딩 모델의 미세조정을 위해 일단 모델을 테스트 해보기]

1.일단모델 다운로드(Bert) 받아서 질답형식(챗봇)으로 잘 기계가 알아먹나 테스트

1. 복잡한 질문들과, 같은 의미이지만 다른 토큰들로 구성되어 있는 질문들도 잘 될수있나 테스트+그리고 이걸 어떻게 파인튜닝해야 잘될지 생각해보기
2. 파인튜닝(peft)으로 잘 되게 해보기(아마 이건 힘들어보인다)

Use-peft?