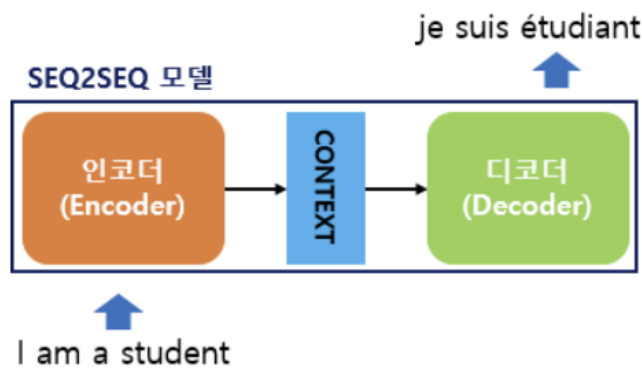


<SENTIBERT>

1. 시퀀스 투 시퀀스 에 관해서

시퀀스 투 시퀀스란, 입력된 시퀀스로 부터 다른 도메인의 시퀀스를 출력하는 다양한 분야에서 사용되는 모델이다. 여기서 시퀀스란 앞뒤 순서가 지켜져야 하는 시-공간적 순서를 말한다. 챗봇과 기계 번역을 대표적 예로 들수있다. (입력 - 질문, 출력 - 대답)

시퀀스 투 시퀀스는 크게 인코더와 디코더라는 두 개의 모듈로 구성되는데, 인코더는 입력 문장의 모든 단어들을 순차적으로 입력받고 마지막에 모든 단어 정보들을 압축하여 하나의 벡터로 만든다. 이것이 컨텍스트 벡터이다. 입력 문장의 정보가 모두 컨텍스트 벡터로 압축되면 인코더는 벡터를 디코더로 전송한다. 디코더는 벡터를 받아서 번역된 단어를 한 개씩 순차적으로 출력한다.



2. Attention 에 관해서

앞서말한 시퀀스 투 시퀀스는 RNN(순환 신경망)을 기반으로 만들어진 모델인데 크게 2가지 문제가 있다. 하나의 고정크기의 벡터에 모든 정보를 압축하려고 하니까 정보손실이 발생하고, RNN의 고질적 문제인 기울기 소실 문제가 존재한다. 기울기 소실 문제란 입력층에 가까운 층들에서 가중치들이 업데이트가 제대로 되지 않아 결국 최적의 모델을 찾을 수 없게 되는 문제를 말한다.

어텐션은 시퀀스 투 시퀀스의 디코더에서 출력 단어를 예측하는 매 시점마다 인코더에서의 전체 입력 문장 중 어느 한 시점에서 예측해야할 단어와 연관이 있는 입력 단어 부분을 좀 더 집중해서 보는 방식이다. 시퀀스 투 시퀀스의 인코더가 입력 시퀀스를 하나의 벡터로 압축하는 과정에서 입력 시퀀스의 정보가 일부 손실된다는 단점이 있었고, 이를 보정하기 위해 어텐션이 사용되었다.

3. BERT에 관해서

먼저 SENTIBERT 나 KEYBERT 에서 동일하게 쓰이는 접미사 BERT에 대해서 알아보자. BERT란 2018년 11월 구글이 공개한 AI 언어 모델로 일부 성능평가에서 인간보다 높은 정확도를 보여 자연어 처리 딥러닝 모델로 주목을 받은 사전 훈련된 모델이다.

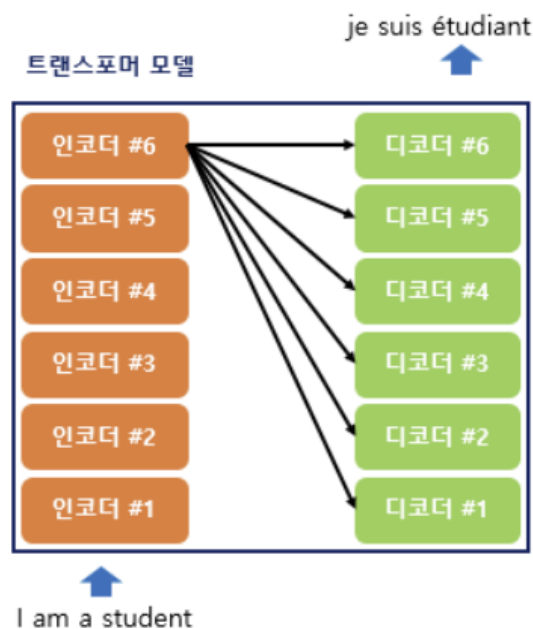
BERT는 모든 출력요소가 모든 입력요소에 연결되고 이들 사이의 가중치가 연결을 기반으로 동적으로 계산되는 딥 러닝 모델인 TRANSFORMER 모델을 기반으로 한다.

BERT의 T가 이

TRANSFORMER의 T다.

3-1. Transformer에 관해서

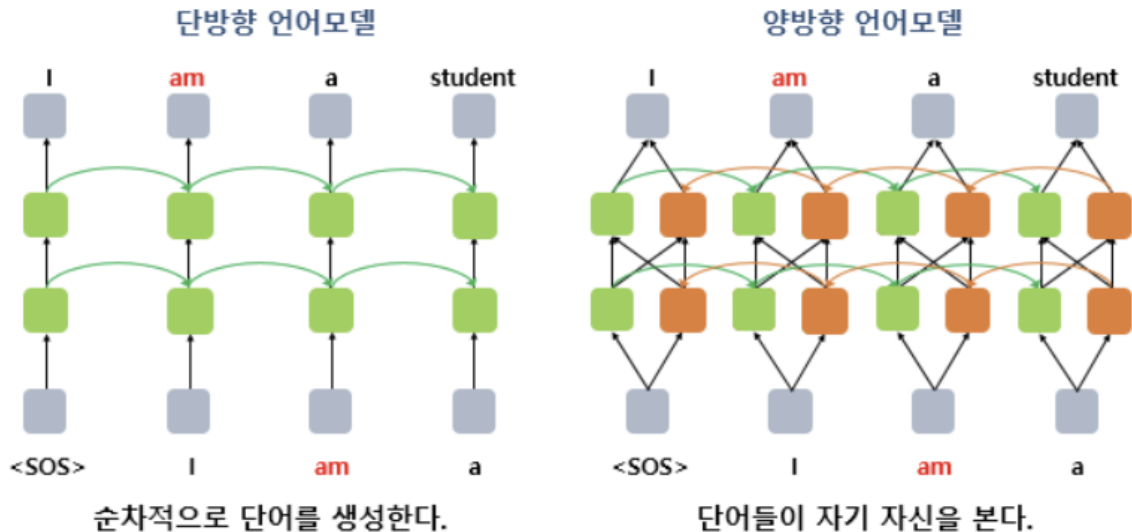
트랜스포머는 시퀀스 투 시퀀스 처럼 RNN을 사용하지 않지만 기존 시퀀스 투 시퀀스처럼 인코더에서 입력 시퀀스를 입력받고 디코더에서 출력 시퀀스를 출력하는 구조를 유지하고 있다.



TRANSFORMER 모델은 문장속 단어와 같은 순차 데이터 내의 관계를 추적하여 맥락과 의미를 학습하는 신경망이다.

역사적으로 언어 모델은 왼쪽에서 오른쪽 또는 오른쪽에서 왼쪽으로 텍스트 입력을 순차적으로만 읽을 수 있었고 동시에 둘 다 할 수는 없었습니다.

하지만 BERT는 한번에 양방향으로 읽는 것이 가능하도록 설계된 모델이다.



BERT의 Pre - Training

= MLM(Masked Language Model) + NSP(NEXT Sentence Prediction)

MLM = 일부 토큰(15%)을 랜덤하게 마스킹, 예측하게 함

NSP = 두 문장의 관계를 파악, 두 문장이 연결되어있는 문장인지 예측. 50%은 연결된 문장
50%은 랜덤한 문장

4. SENTIBERT에 관해서

sentiBERT 연구 논문을 조회해본 결과

A Transferable Transformer-Based Architecture for Compositional Sentiment Semantics

직역하자면 트랜스포머 모델을 기반으로 한 구성 감정 의미론이라는 뜻이다. 여기서 구성이란 문장(sentence)을 말한다. 알기 쉽게 풀어서 설명하자면 문장의 담긴 감정과 정서, 의미를 Transformer 모델을 기반으로 알아내는 NLP(자연어 처리) 기법이라고 보면 될 것이다.

sentiBERT는 문장 수준에 효과적인 기법이다.

BERT + binary constituency parse tree

sentiBERT란 BERT의 변형으로 효과적으로 감정과 의미를 파악한다. sentiBERT는 이진 구문 분석 트리(binary constituency parse tree)를 통해 문맥화된 표현 속 의미와 구성을 파악한다.

frenetic but not really funny (열광적이지만 정말 재미있지 않은) 예시로 들었을 때

but은 문장의 sentiment(감정)를 바꾸고 not은 really funny의 sentiment를 바꾼다.

일반적으로 문장은 단어(토큰)와 구절의 의미와 결합되는 방식에 의해 의미가 달라진다.

sentiBERT를 사용하는 이유 :

문장 단위로 나누어 문장의 앞, 뒤 맥락을 파악하여 해당 문장이 가진

감정(좋고 싫음), 의미를 파악해 내기 위해 사용할 수 있다.

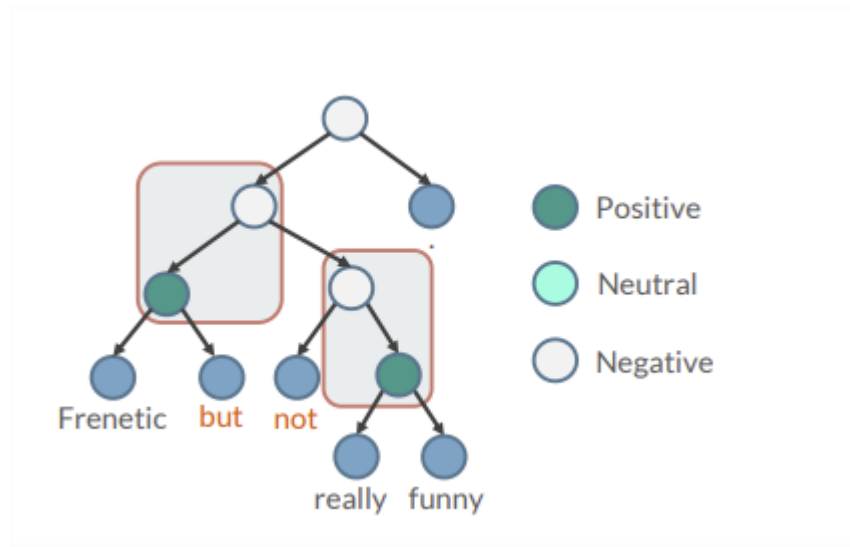
-> 리뷰가 긍정, 부정인가를 판별하기 위해 필요한 자연어 처리 기법.

sentiBERT

1) sentiBERT

sentiBERT는 BERT에서 파생 된 자연어 처리 기법으로 효과적으로 문장에 담긴 감정과 의미를 알아내기 위해 사용한다.

sentiBERT의 자연어 처리 방식은 BERT와 이진 분석 트리로 작동된다.



논문에서 예시로든 “frenetic but not really funny” 를 보면

해당 문장은 문장 속에 담긴 감정을 변화시키는 “but” 으로 연결된 문장이다.

그리고 “not” 또한 “really funny”의 감정을 변화시킨다.

이 처럼 문장 속 감정은 토큰(단어)과 문구의 의미(=not)에 따라 결정되고

구문적으로 결합되는 방식(=but)에 따라 다르게 표현된다.

선행 연구로 RNN (순환 신경망)을 통해 문장 속 감정, 의미를 모델링하려고

했다. 하지만 RNN을 사용하여 모델링하는 방식은 문맥의 풍부한 연관성을

간과하는 모델링 방식이다.

23.01.25 (수) study

- 왜 우리는 **BERT**를 사용해야하는가?

“enabling transfer learning with large language models that can capture complex textual patterns”

BERT는 복잡한 텍스트 패턴을 캡처할 수 있는 대규모 언어 모델로

전이 학습을 가능하게 한다.

++ GPT-3 같은 경우엔 비용도 발생 할 뿐더러 **VRAM**용량이 매우 커서 개인 컴퓨터로는 돌릴 수 없다.

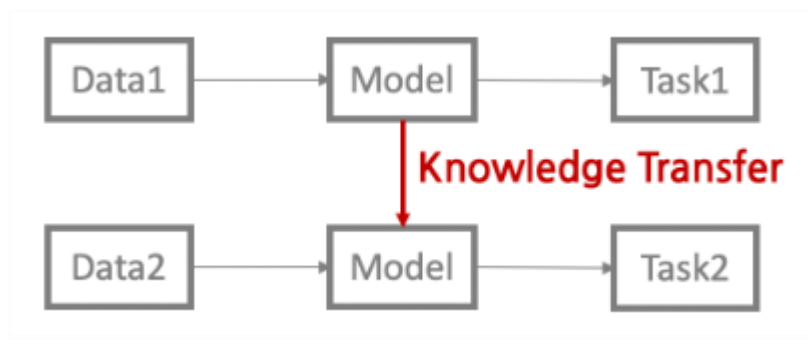
- 전이 학습(**transfer learning**)이란 무엇인가?

특정 작업에서 사용하여 학습한 모델을 다른 작업 수행에 재사용하는 기법을 가리킨다. 비유하자면 사람이 새로운 지식을 배울 때 그가 평생 쌓아왔던 지식을 요긴하게 다시 써먹는 것과 같음.

전이 학습을 적용하면 ***기존**보다 모델의 학습 속도가 빨라지고 새로운 작업을 더 잘 수행하는 장점이 있음. **BERT**나 **GPT**가 이 기법을 사용 중.

*여기서 기존 모델이란 전이 학습 적용 이전 모델로 이전 모델은 작업을 처음부터 학습했음.

- 업스트림 태스크와 다운스트림 태스크



위 그림의 **task1**은 업스트림 태스크이고 **task2**는 다운스트림 태스크이다.

업스트림 태스크를 학습하는 과정을 프리트레인이라고 한다. (**pre-train**)

다운스트림 태스크를 수행하기에 앞서 학습하는 과정이기 때문.

업스트림 태스크는 다음 단어 맞추기와 빈칸 채우기가 있는데 **BERT**, **GPT** 계열 모델이 이 태스크로 프리트레인을 수행한다.

BERT의 Pre-Training 인 **MLM**과 **NSP** = 업 스트림 태스크

다운스트림 태스크는 문서 분류, 개체명 인식 등 우리가 해결하고 싶은 자연어 처리의 구체적인 문제들이다. 업스트림 태스크를 **pre-training** 한 이유는 바로 이 다운스트림 태스크를 잘 하기 위해서다.

transfer learning = 업스트림 태스크(사전학습작업) + 다운스트림 태스크(사전학습을 기반으로 문제해결작업)

- transformer란?

RNN의 순차 처리 구조는 왜 문제가 있을까?

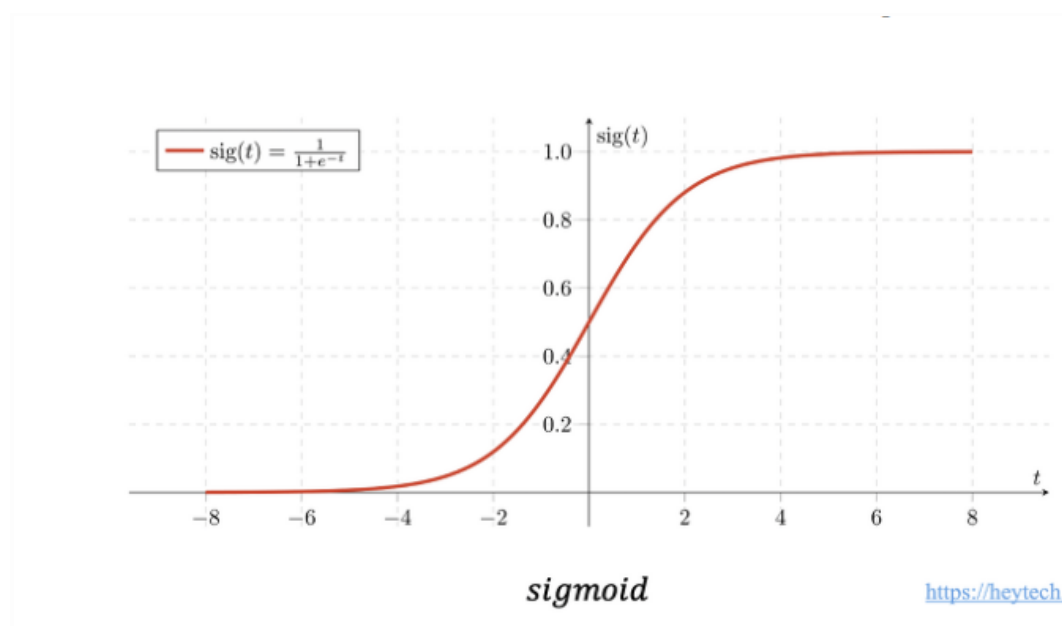
“반바지를 입고 아이스크림을 들고 있던 아이는 그것을 금방 다 먹었다.”
순차적으로 처리한다면 프로그램은 “그것”이 반바지인지 아이스크림인지 모름.
뒤에 “먹었다”라는 표현을 알아야 아이스크림이라고 유추 할 수 있지만 단 방향이기 때문에 “먹었다”라는 표현이 **input** 될 땐 이미 “그것”은 **output**되었기 때문.
앞 뒤로 분석하는 양방향 RNN도 등장했지만 구조적으로 앞부분의 “아이스크림”과 뒷부분의 “먹는다”를 서로 동시에 참조하지 못하기 때문에 해결방안이 되지 못했다.
해결방안으로 **Attention** 매커니즘이 사용됐는데 **Attention**은 번역을 할 때 여러 부분을 한꺼번에 참고하는 메커니즘 이다. 이제 **Attention**만 있으면 되기 때문에 RNN은 사용이 안되고 **Attention**과 *워드 임베딩 벡터, 그리고 단어 순서를 위한 **positional encoding**을 추가하게된다.

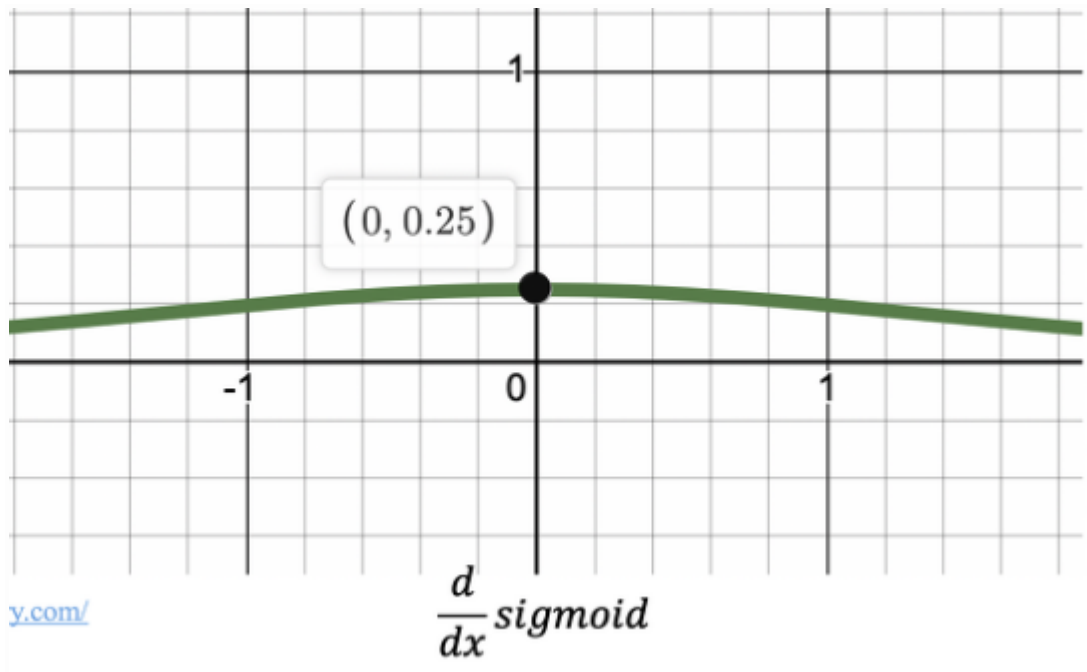
RNN의 단점 : 고정된 크기의 벡터에 모든 정보 압축 -> 정보손실 발생

RNN 의 고질적 문제인 기울기 소실 문제 -> 최적의 모델 찾을 수 없게됨

기울기소실 -> 활성화 함수의 기울기와 관련이 있음

활성화 함수 중 **sigmoid**로 예를 들면, x값이 크거나 작아짐에 따라 기울기가 거의 0에 수렴하게 됨. 이때 컴퓨터가 계산할 때 정확한 값이 아닌 근사값으로 계산해야 되므로 학습 오차가 발생하게 되어 최적의 모델을 찾을 수 없게 되는것.





-> **positional encodding** : 트랜스포머는 단어를 순차적으로 받는 방식이 아니기 때문에 단어의 위치 정보를 다른 방법으로 알려 줄 필요가 있음. 이때 사용.

sin cos 함수를 합성한 위치를 벡터로 표현.

단어의 위치를 신호로 표현하는 것이 성능에 효과적이기 때문

*워드임베딩 = 단어를 벡터로 표현하는 방법

- 어텐션이란 ?

트랜스포머의 기반이 되는 매커니즘.

어텐션의 기본 아이디어는 디코더에서 출력 단어를 예측하는 매 시점마다 인코더에서 전체 입력 문자를 다시 한 번 참고하는것. 해당 시점에서 예측해야 할 단어와 연관이 있는 입력 단어 부분을 좀 더 집중적으로 보게 됨.

논문에 나온 related work :

- sentiment analysis (감정분석)

이진 탐색 트리를 활용한 트랜스포머 기반 신경망 모델을 통하여 문장 속 감정 캡처를 한다고 함.

- semantic compositionality (의미 구성)

해당 작업은 sentiment analysis(감정분석), semantic relatedness(의미론적 연관성), capture sememe knowledge(의미론적 지식같은)에 응용된다고 함. 의미 구성이 더 잘 캡처 될 수 있도록 구문 구조는 문맥 속 표현과 결합될 수 있다는 것을 증명하겠다고함.