

A row of white puzzle pieces is held horizontally by two hands, one on the left and one on the right. The puzzle pieces are interlocked and form a continuous line. The background is solid black. A semi-transparent white rectangular box is centered over the puzzle pieces, containing the title and subtitle text.

# NLU – Semantic textual Similarity

WANTED x CODE STATES 5팀 최지나

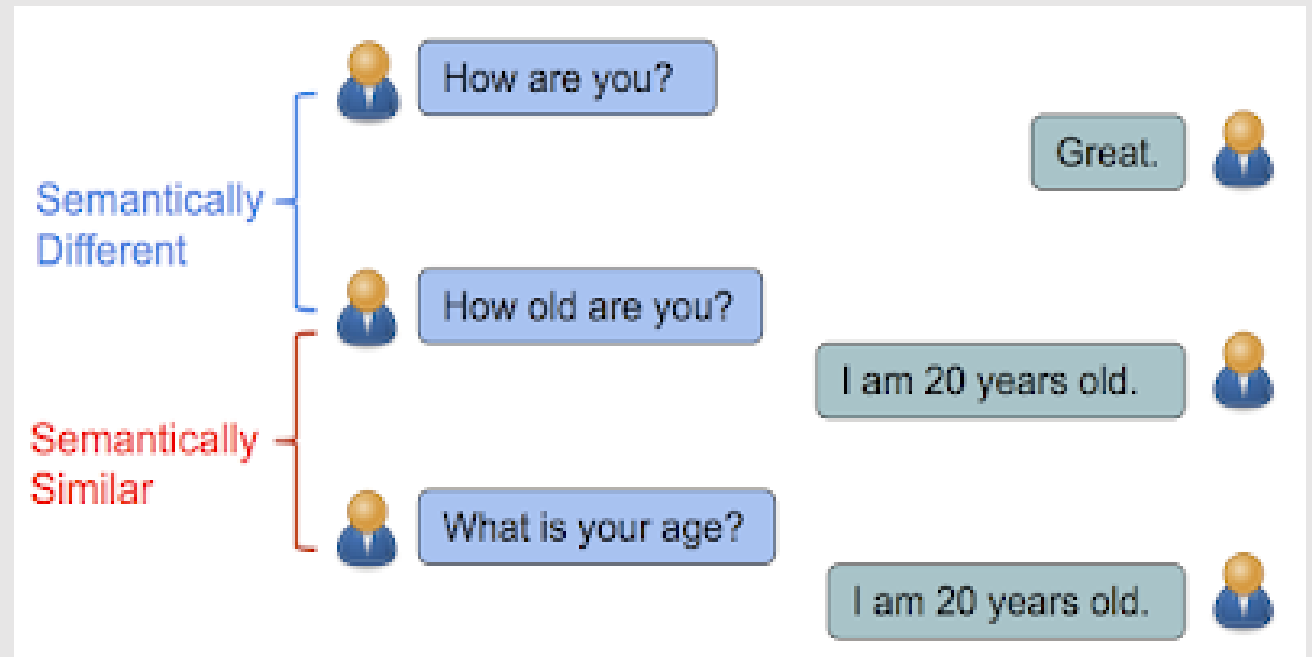
# 1. 프로젝트 주제 및 담당 역할

## 1) 과제 목표

- 한국어 문장의 유사도 분석 모델 훈련 및 서비스화
- 의미적 텍스트 유사도(Semantic Textual Similarity) : 두 문장 사이의 의미적 동등성의 정도를 측정.
- 학습 데이터 셋을 사용하여, 의미적 텍스트 유사도 모델을 훈련하는 것을 목표로 합니다.
- 두 개의 한국어 문장을 입력 받아 두 문장의 의미적 유사도를 출력하는 모델 생성

## 2) 담당 역할

- 논문 리서치, 모델링, 하이퍼 파라미터 튜닝



## 2. 학습 방향

### 1) 개인 학습

- word-transformer 모델인 BERT에 대한 강의 수강
- 과제 해결을 위한 sentence-transformer model에 대한 공부 및 코드 구현

### 2) 공동 학습

- Discord와 Notion을 통해 유사도 측정을 위한 모델과 방법론에 대한 조사
- Git-hub를 통해 .ipynb 파일로 돌린 결과들을 직관적으로 확인할 수 있게 공유

### 3. 모델 선정 이유

## Sentence-BERT(이하 'S-BERT')

- 모델 관련 논문 : [Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks](#)

- 선정 이유 : 모델의 효율적 연산을 위해서

- BERT를 이용한 STS 계산의 한계점

BERT는 기본적으로 Cross-Encoder 모델임.

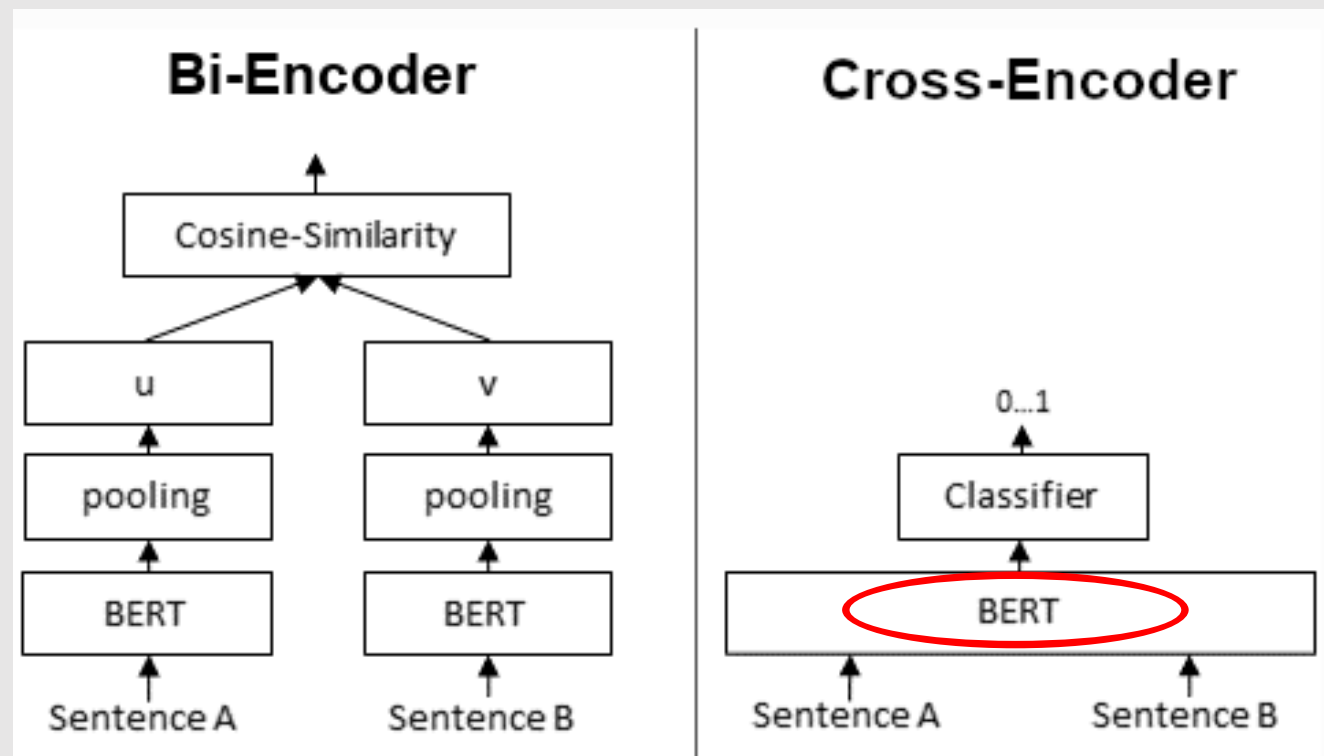
두 문장의 유사도를 비교하려면 Bi-Encoder의 도식화처럼 동시에 transformer network를 통과하여 매번 새롭게 변환된 임베딩 값으로 계산되어야 함.

→ 다수의 비교 쌍이 존재할 경우, 연산량 급증

- BERT에서의 기존의 한계를 극복하기 위한 방법

- 1) 문장의 각 토큰 output\_layer의 hidden-states 평균 계산
- 2) Context vector의 의미로서 [CLS] 토큰의 hidden\_state를 사용함.

※ 그러나 연산량이 많고, 정확도가 떨어짐.



출처 : <https://www.sbert.net/examples/applications/cross-encoder/README.html>

## 4. 하이퍼 파라미터 튜닝 결과

- Pretrained 모델(max\_seq\_length:128, batch\_size=8, num\_epochs=5)




- 'KLUE-RoBERTa-large' : 0.88814
- 'KLUE-RoBERTa-base' : 0.88881

- 'KLUE-RoBERTa-base' (max\_seq\_length:128, batch\_size=8, num\_epochs=5)

Structure	activation	Pearson
word_embedding-Pooling-Dense	Tanh	0.83477
word_embedding-Pooling-Dense	ReLU	0.71662
word_embedding-Pooling	X	0.88881

- 'KLUE-RoBERTa-base' word\_embedding-Pooling

max_seq_length	batch_size	num_epochs	Pearson
128	8	5	0.88881
64	8	5	0.89338
32	8	5	0.88591
64	16	5	0.89320
64	32	5	0.88473

#	Team	Model	Description	YNAT	KLUE-STS	
				F1 	R <sup>P</sup> 	F1 
1	KLUE-team	KLUE-BERT-base	More	85.73	90.85	82.84
2	KLUE-team	KLUE-RoBERTa-large	More	85.69	93.35	86.63

- 최고의 조합

'KLUE-RoBERTa-base',

word\_embedding-Pooling,

max\_seq\_length = 64, batch\_size = 8, num\_epochs= 5

## 5. 훈련 과정

- 특수문자, 띄어쓰기, html 태그, 영어 소문자화 등 Dirty data 정제
- Real-label을 0~1 사이로 정규화

Data Preprocessing

Make a Dataset

'sentence\_transformers.readers'의  
InputExample 이용하여  
{ text :(sentence1, sentence2), label : score}  
형태의 Dataset 생성

Train Model

Evaluate Model

훈련된 모델을 이용하여 test dataset을 넣어 평가.

튜닝 결과 중 가장 좋은 조건으로 모델 훈련

## 5. 최종 결과 분석

### 1) Real\_Label의 경우

- 0~5점 사이이기 때문에 정규화 작업을 하여 모델이 조금이라도 쉽게 예측할 수 있도록 작업하였으나, f1-score에서 다소 낮은 점수를 보였다고 생각.
- F1-Score가 쓰이는 경우에 대해서 알아보니, 분류 클래스 간의 데이터가 심각할 때 사용한다고 하였다. 그러나 label을 기준으로 데이터를 group-by 하였을 때 일반적인 데이터에 비해 균일한 데이터로 판단.
- 따라서 F1-score 뿐만 아니라 Accuracy, Precision, Recall score를 추가적으로 확인이 필요함.

### 2) Binary\_Label의 경우

- Binary라는 형식의 특성상, 0과 1 중 하나이기 때문에 성능이 좋을 것으로 처음부터 예상함.
- 모델을 훈련할 때, dataset의 label을 binary로 하고, 그에 따른 loss function, evaluator를 사용하였을 때 같은 조건에서의 real\_label 모델에서의 성능보다 좋은 성능을 보임.
- 위 경우에는 피어슨 상관계수가 상당히 높았지만, 이번 경우에는 F1 score가 좋은 점수를 보이고 있다. 그러나 피어슨 상관계수도 F1 score와 크게 차이가 나지 않는 편이기 때문에 문장의 유사도를 판단할 때는 가능한, binary로 label을 매기는 것이 좋은 결과들을 추출할 것이라 예상함.