MTEB: Massive Text Embedding Benchmark

Name

Sangho Daniel Kim

Study Group

2024/05/07

TRAIN AND TEST

Contents

TRAIN AND TEST

Why Benchmarks?

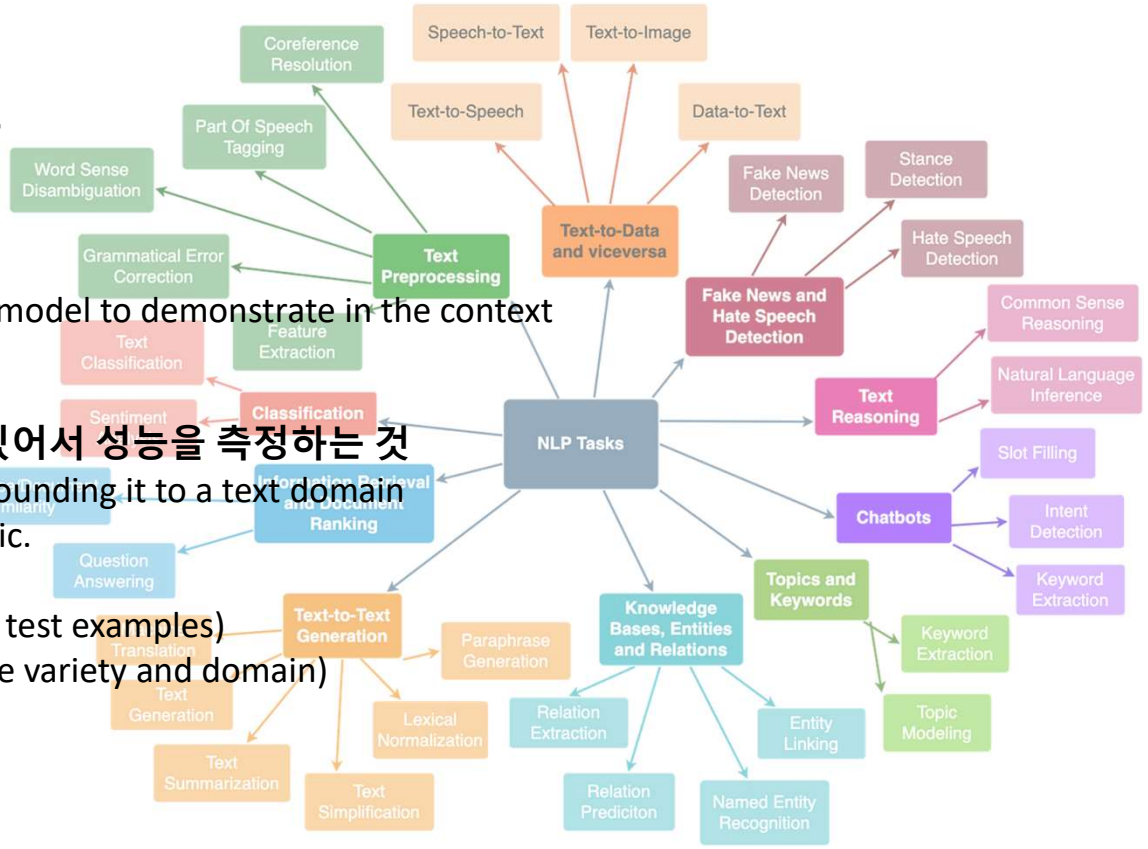
Definition

Bench·mark : 특정 기준을 가지고 대상을 평가하고 비교하다
to evaluate or check by comparison with a standard

Task : Model이 특정 input에 대해 구현했으면 하는 Skill
A task is a language-related skill or competency that we want a model to demonstrate in the context of a specific input–output format OR abstract skill specification

Benchmarks : 특정 Task에 대한 Performance를 실현하는데 있어서 성능을 측정하는 것
A benchmark attempts to evaluate performance on a task by grounding it to a text domain and instantiating it with a concrete dataset and evaluation metric.

- **데이터셋**: a specific sample of passages and questions (set of test examples)
- **다언어** : from the English personal narrative domain (language variety and domain)
- **여러 척도** : test using an accuracy metric (concrete metric)



TRAIN AND TEST

Why Benchmarks?

Reasons for its need

1. 상대적 성능을 일관된 데이터셋과 평가 지표로 정량적 측정

2. 실제 세계 데이터에서 얼마나 잘 동작하는지를 확인

3. 모델의 강점과 약점을 파악

4. 어떤 모델이 해당 Task를 얼마나 잘 수행하는지를 평가, 공인된 신뢰성 부여

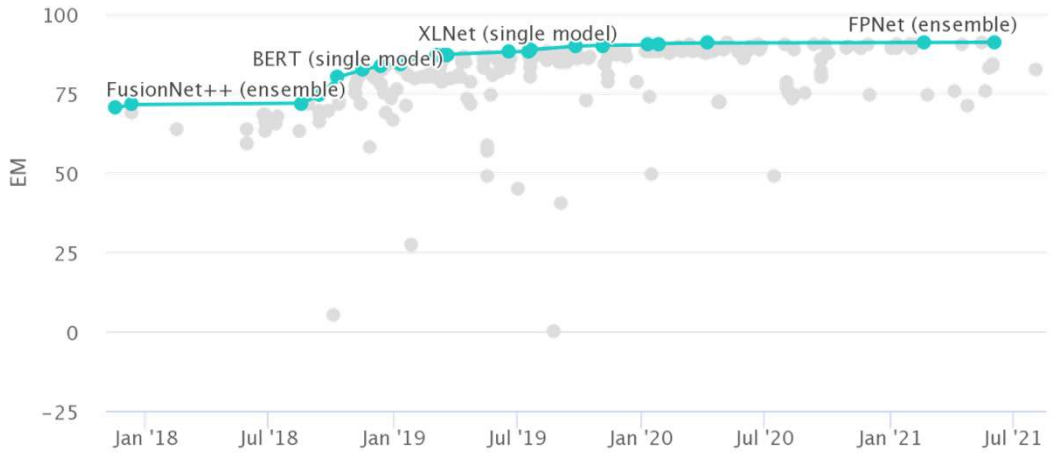
TRAIN AND TEST

Why Benchmarks?

Why do we need more?

기술이 발전함에 따라(Transfer Learning 등)을 통해 **더더욱 강력해진 NLP Model들을 평가하고 비교할 기준이 부족해짐**
- Recent models "have outpaced the benchmarks to test for them" (AI Index Report 2021)

traditional practices for evaluating performance of NLP models, **using a single metric** such as accuracy or BLEU, relying on static benchmarks and abstract task formulations **no longer work** as well in light of models' surprisingly robust *superficial* natural language understanding ability.



TRAIN AND TEST

Why Benchmarks?

Problem

문제점1 : 현재까지는 하나의 Task 에서 비롯된 한정적인 크기의 Dataset 들만 이용하여 평가했음

Text embeddings are commonly evaluated on a small set of datasets from a single task **not covering their possible applications to other tasks.** This makes progress in the field difficult to track, as various models are constantly being proposed without proper evaluation.

최근 개발되고 있는 모델들은 다양한 Task 영역까지 건드리는 만큼, 다양한 기준에서 성능 측정을 할 필요가 있음
However, the evaluation regime of current text embedding models rarely covers the breadth of their possible use cases.

그렇다면 Text Embedding Model이 적용될 수 있는 여러 영역에서의 성능은 어떻게 측정할 수 있을까?

TRAIN AND TEST

Why Benchmarks?

Problem

문제점2 : 다양한 task에 대한 Embedding Method들의 성능을 평가하는 것은 여러 Evaluation Pipeline을 요구

Pre-processing 또는 hyperparameter과 같은 **이미 구현된 값들이 결과에 영향**을 미칠 수 있음

특정한 task에서의 성능 개선이 단순히 **유리한 평가 파이프라인에서 비롯되는지 여부가 불분명**

실제 산업에서 적용이 된다면 **평가를 다시 하게 되는 문제**가 생김

Further, evaluating embedding methods on many tasks requires implementing multiple evaluation pipelines. Implementation details like pre-processing or hyperparameters may influence the results making it unclear whether performance improvements simply come from a favorable evaluation pipeline.

기존의 Benchmark과 달리 전반적인 평가가 가능한 Pipeline을 지닌 새로운 Benchmark 요구

TRAIN AND TEST

Introduction

Introduction

"무엇을 좋아할지 몰라 일단 다 넣어봤습니다"

MTEB: Massive Text Embedding Benchmark

여러 task에 대해 적용 가능한 Ultimate text embedding model은 무엇일까?

A gateway to finding universal text embeddings applicable to a variety of tasks

Niklas : "Project to find the best text embedding model"

TRAIN AND TEST

Introduction

Introduction

목표 : Model 들이 다양한 Embedding Task에 대해 얼마만큼의 성능을 보여주느냐를 측정
The Massive Text Embedding Benchmark (MTEB) aims to provide clarity on how models perform on a variety of embedding tasks

총 58개의 datasets, 112개의 언어 지원, 8개의embedding tasks에 대한 전반적인 평가

1. Bitext mining
2. Classification
3. Clustering
4. Pair classification
5. Reranking
6. Retrieval
7. STS (Semantic Textual Similarity)
8. Summarization

"이것으로 각 모델들의 Weaknesses와 Strengths를 측정하겠다."

TRAIN AND TEST

Related Works

2.1 Benchmarks

이미 다양한 벤치마크들이 실존함

1. (Super)GLUE (Wang et al., 2018, 2019)
2. Big-BENCH (Srivastava et al., 2022)
3. SemEval datasets (Agirre et al., 2012, 2013, 2014, 2015, 2016)

MTEB 만의 특이점

1. 서로 다른 Embedding tasks에 쓰이는 dataset들을 하나의 일반적인 평가 Framework로 통합
2. 가장 최신의 벤치마크인 SemEval, BEIR 데이터셋까지 통합

= Text embedding model 들의 전반적인 performance 측정을 가능하게 함

MTEB unifies datasets from different embedding tasks into a common, accessible evaluation framework. MTEB incorporates SemEval datasets (STS11 – STS22) and BEIR alongside a variety of other datasets from various tasks to provide a holistic performance review of text embedding models.

TRAIN AND TEST

Related Works

2.2 Embedding Models

- **GloVe (Pennington et al., 2014)** – Context Awareness 부족 (Word Embedding Model)
- **Transformers (Vaswani et al., 2017)** –Self-attention을 통한 Context Awareness 부여
- **BERT (Devlin et al., 2018)** – Transformer architecture 사용, large-scale self-supervised pre-training
- **SBERT (Reimers and Gurevych, 2019)** – Transformer에 추가적인 fine-tuning이 embedding performance 향상 유도

저자들의 예측

1. 근래의 Fine-tuned Embedding Models들은 Contrastive Loss를 구하여 긍정/부정 짝에 대한 Supervised fine-tuning 이용
2. 현존하는 pre-trained Transformers (Wolf et al., 2020)들만으로도 앞으로 다양한 Text Embedding Model들이 등장할 것

Word Embedding과 Transformer 적용 모델들(Context-aware)을 동시에 측정 가능하도록 설계

TRAIN AND TEST

The MTEB Benchmark

3.1 Desiderata

다양성 Diversity

8 tasks * 각각 최대 15개의 dataset 부여 = 총 58개의 datasets
10개의 multilingual dataset, 112개 언어 제공
Sentence-level / Paragraph-level datasets 포함

간결성 Simplicity

API 제공을 통해 다양한 model들에 대한 적용 및 평가를 쉽게 함

확장성 Extensibility

계속되는 벤치마크 업데이트(새로운 task 및 dataset)
현존하는 task : 파일 형식으로 Hugging Face dataset에 업로드
새로운 taks : Task interface / Evaluator 필요

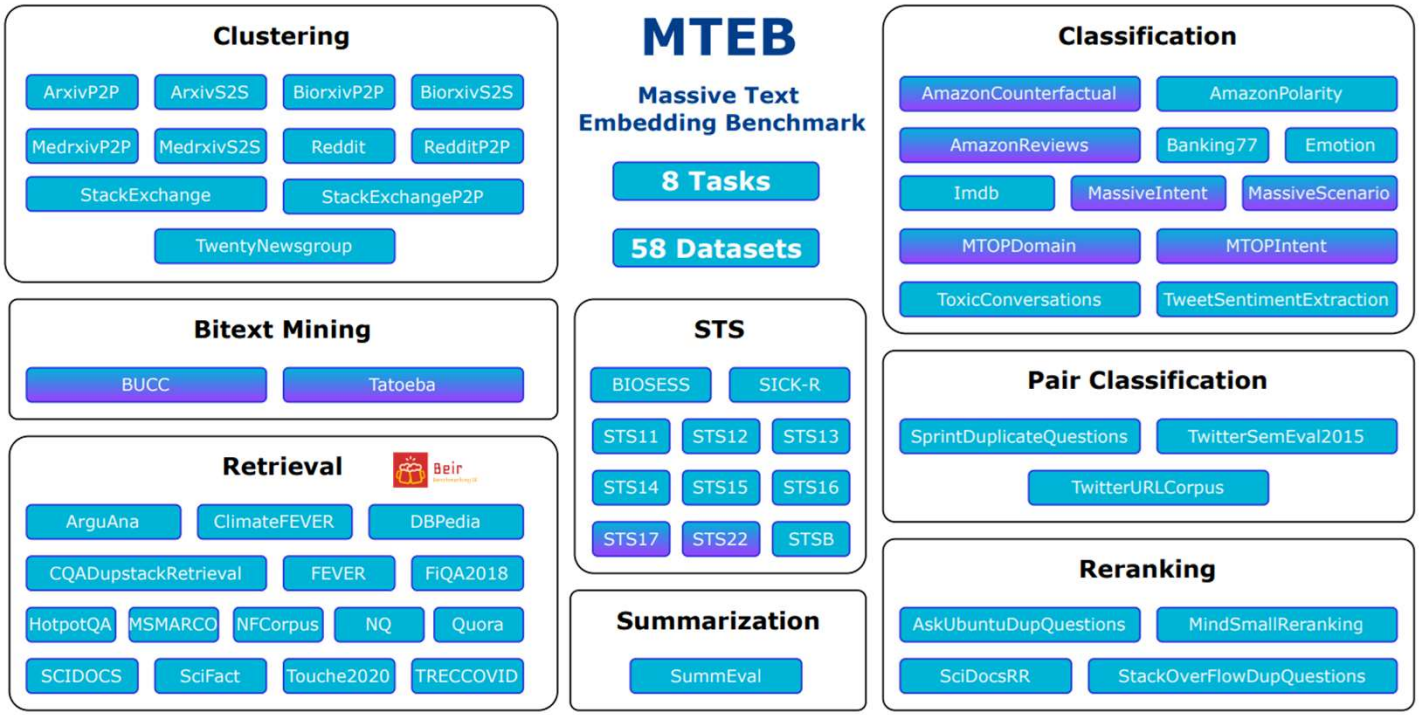
공유성 Reproducibility

결과를 JSON 파일 형식으로 저장하여 공유를 간편하게 함

TRAIN AND TEST

The MTEB Benchmark

3.2 Tasks and Evaluation



TRAIN AND TEST

The MTEB Benchmark

3.2 Tasks and Evaluation

Bitext Mining	다른 언어로 구성된 두 세트의 문장 입력, **첫 세트의 문장에 대응되는 문장을 타 세트에서 선택** (Main Metric : 코사인 유사도, F1)
Classification	Train/Test set을 모델과 함께 입력, Train set로 **로지스틱 회귀 분류기 훈련 후 Test set 평가** (Main Metric : Accuracy with average precision, F1)
Clustering	문장과 문단의 집합을 입력, **유의미한 그룹으로 집합 분류** label 수를 군집의 개수(K)로 설정한 K-means model로 embedded text로 학습, V-measure을 통해 학습
Pair-Classification	Text 쌍 입력 후 **Label을 할당, Embedded 완료된 값들 사이의 거리 계산** (Main Metric : Average Precision Score based on Cosine Similarity)
Reranking	Query와 유/무관한 reference text 입력, **Query 문과의 결과 관련도**에 따른 ranking 진행 각 쿼리와 전체 쿼리의 평균을 통해 최종 ranking 결정 (Main Metric : MAP)
Retrieval	**정보 검색**, 문서 집합과 Query, Query 관련 문서를 집합에서 찾기 위한 매핑 입력 제공된 모델은 Query와 문서 집합들을 embedding 한 후, 유사도 측정 (Main Metric : nDGC@10)
Semantic Textual Similarity	두 가지 문장을 주고 **각 문장을 embedding 한 값**에 대한 유사도 측정 (Main Metric : Spearman correlation based on cosine similarity)
Summarization	Human-based 요약과 machine-generated 요약이 제공, **각 요약의 embedding 값에 대한 유사도 측정** (Main Metric : Spearman correlation based on cosine similarity)

TRAIN AND TEST

The MTEB Benchmark

Datasets

MTEB에 들어간 Dataset 들에 대한 유사도 측정값

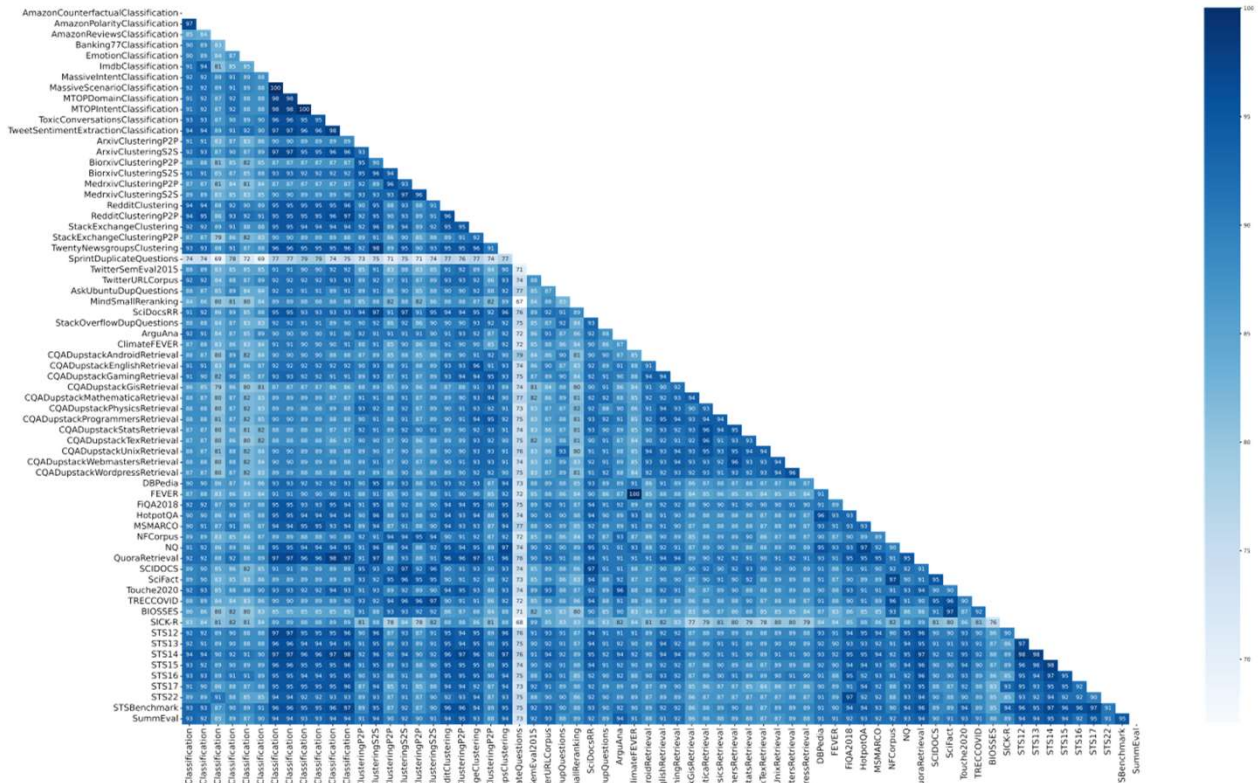


Figure 2: Similarity of MTEB datasets.
We use the best model on MTEB STS (ST5-XXL, see Table 1) to embed 100 samples for each dataset. Cosine similarities between the averaged embeddings are computed and visualized.

TRAIN AND TEST

The MTEB Benchmark

Datasets

MTEB의 다양성 확보를 위해, Dataset 내부 다양한 길이의 text를 포함

Sentence to Sentence (S2S)	• 문장과 다른 문장을 비교 • 모든 STS task들이 S2S의 예시 (두 문장 간의 유사도 측정의 관점)
Paragraph to Paragraph (P2P)	• 문단을 다른 문단과 비교 • MTEB는 Input 길이 제한이 없음 – 모델들에게 문단 자르기 맡김 • 몇몇 Clustering task들이 S2S P2P task를 포함 (S2S는 title들만 비교 / P2P는 title과 content 모두 반영)
Sentence to Paragraph (S2P)	• Retrieval Dataset 중 몇 개 포함 • Query가 Single Sentence이고 Document가 긴 Paragraph에 해당

TRAIN AND TEST

Results

4.1 Models

Self-Supervised Methods	Transformer-based	**BERT(Devlin et al., 2018)** Mean-pooling(sequence 길이의 평균을 구하는 방법)을 통해 text embedding 직접 생성 **SimCSE-Unsup(Gao et al., 2021b)** (BERT 기반, 추가적인 자기 지도 학습)	
	Non-Transformer	**Komninos (Komninos and Manandhar, 2016)** and **GloVE (Pennington et al., 2014)** 단어를 벡터로 바로 변환하여 embedding 값들은 문단 이해도(Context Awareness)가 부족	
Supervised Methods	Transformer encoder methods	**coCondenser (Gao and Callan, 2021)** **Contriever (Izacard et al., 2021)** **LaBSE (Feng et al., 2020)** **SimCSE-BERT-sup (Gao et al., 2021b)**	**SPECTER (Cohan et al., 2020a)** **GTR (Ni et al., 2021b)** **ST5 (Ni et al., 2021a)** **MPNet, MiniLM**
	Transformer decoder methods	**SGPT BiEncoders (Muennighoff, 2022)** **SGPT-nli models** **SGPT-msmarco models** **cpt-text (Neelakantan et al., 2022)**	
	Non-Transformer	**LASER (Heffernan et al., 2022)** - 유일한 Context-aware non-Transformer Model (LSTM)	

TRAIN AND TEST

Results

4.2 Analysis

여러 영문 task들에 대해 모두 최고의 성능을 보여준 Model은 없었다.

There is considerable variability between tasks!
No model claims the state-of-the-art in all seven English tasks.

Self-Supervised Method 기반 모델과 Supervised Method 기반 모델의 차이가 확연하다.

향상된 성능을 위해서는
추가적인 supervised fine-tuning이 필요

Num. Datasets (→)	Class. 12	Clust. 11	PairClass. 3	Rerank. 4	Retr. 15	STS 10	Summ. 1	Avg. 56
Self-supervised methods								
Glove	57.29	27.73	70.92	43.29	21.62	61.85	28.87	41.97
Komninos	57.65	26.57	72.94	44.75	21.22	62.47	30.49	42.06
BERT	61.66	30.12	56.33	43.44	10.59	54.36	29.82	38.33
SimCSE-BERT-unsup	62.50	29.04	70.33	46.47	20.29	74.33	31.15	45.45
Supervised methods								
SimCSE-BERT-sup	67.32	33.43	73.68	47.54	21.82	79.12	23.31	48.72
coCondenser-msmarco	64.71	37.64	81.74	51.84	32.96	76.47	29.50	52.35
Contriever	66.68	41.10	82.53	53.14	41.88	76.51	30.36	56.00
SPECTER	52.37	34.06	61.37	48.10	15.88	61.02	27.66	40.28
LaBSE	62.71	29.55	78.87	48.42	18.99	70.80	31.05	45.21
LASER2	53.65	15.28	68.86	41.44	7.93	55.32	26.80	33.63
MiniLM-L6	63.06	42.35	82.37	58.04	41.95	78.90	30.81	56.26
MiniLM-L12	63.21	41.81	82.41	58.44	42.69	79.80	27.90	56.53
MiniLM-L12-multilingual	64.30	37.14	78.45	53.62	32.45	78.92	30.67	52.44
MPNet	65.07	43.69	83.04	**59.36**	43.81	80.28	27.49	57.78
MPNet-multilingual	67.91	38.40	80.81	53.80	35.34	80.73	**31.57**	54.71
OpenAI Ada Similarity	70.44	37.52	76.86	49.02	18.36	78.60	26.94	49.52
SGPT-125M-nli	61.46	30.95	71.78	47.56	20.90	74.71	30.26	45.97
SGPT-5.8B-nli	70.14	36.98	77.03	52.33	32.34	80.53	30.38	53.74
SGPT-125M-msmarco	60.72	35.79	75.23	50.58	37.04	73.41	28.90	51.23
SGPT-1.3B-msmarco	66.52	39.92	79.58	54.00	44.49	75.74	25.44	56.11
SGPT-2.7B-msmarco	67.13	39.83	80.65	54.67	46.54	76.83	27.87	57.12
SGPT-5.8B-msmarco	68.13	40.35	82.00	56.56	**50.25**	78.10	24.75	58.81
SGPT-BLOOM-7.1B-msmarco	66.19	38.93	81.90	55.65	48.21	77.74	24.99	57.44
GTR-Base	65.25	38.63	83.85	54.23	44.67	77.07	29.67	56.19
GTR-Large	67.14	41.60	85.33	55.36	47.42	78.19	29.50	58.28
GTR-XL	67.11	41.51	**86.13**	55.96	47.96	77.80	30.21	58.42
GTR-XXL	67.41	42.42	86.12	56.65	48.48	78.38	30.64	58.97
ST5-Base	69.81	40.21	85.17	53.09	33.63	81.14	31.39	55.27
ST5-Large	72.31	41.65	84.97	54.00	36.71	81.83	29.64	57.06
ST5-XL	72.84	42.34	86.06	54.71	38.47	81.66	29.91	57.87
ST5-XXL	**73.42**	**43.71**	85.06	56.43	42.24	**82.63**	30.08	**59.51**

Table 1: Average of the main metric (see Section 3.2) per task per model on MTEB English subsets.

TRAIN AND TEST

Results

4.2 Analysis

Classification
- ST5 모델들이 state-of-the-art 성능을 보이고 있음

Clustering
- MPNET과 ST5-XXL

Pair Classification
- GTR-XL과 GTR-XXL

Reranking
- MPNet과 MiniLM

Retrieval
- SGPT-5.8B-msmarco

STS & Summarization
- ST5-XXL

Q. STS에 최적화된 모델들(SimCSE, ST5, SGPT-nli)이 Retrieval에서 저조한 성능을 보이는 근본적인 이유는 무엇일까?

Num. Datasets (→)	Class. 12	Clust. 11	PairClass. 3	Rerank. 4	Retr. 15	STS 10	Summ. 1	Avg. 56
Self-supervised methods								
Glove	57.29	27.73	70.92	43.29	21.62	61.85	28.87	41.97
Komninos	57.65	26.57	72.94	44.75	21.22	62.47	30.49	42.06
BERT	61.66	30.12	56.33	43.44	10.59	54.36	29.82	38.33
SimCSE-BERT-unsup	62.50	29.04	70.33	46.47	20.29	74.33	31.15	45.45
Supervised methods								
SimCSE-BERT-sup	67.32	33.43	73.68	47.54	21.82	79.12	23.31	48.72
coCondenser-msmarco	64.71	37.64	81.74	51.84	32.96	76.47	29.50	52.35
Contriever	66.68	41.10	82.53	53.14	41.88	76.51	30.36	56.00
SPECTER	52.37	34.06	61.37	48.10	15.88	61.02	27.66	40.28
LaBSE	62.71	29.55	78.87	48.42	18.99	70.80	31.05	45.21
LASER2	53.65	15.28	68.86	41.44	7.93	55.32	26.80	33.63
MiniLM-L6	63.06	42.35	82.37	58.04	41.95	78.90	30.81	56.26
MiniLM-L12	63.21	41.81	82.41	58.44	42.69	79.80	27.90	56.53
MiniLM-L12-multilingual	64.30	37.14	78.45	53.62	32.45	78.92	30.67	52.44
MPNet	65.07	43.69	83.04	**59.36**	43.81	80.28	27.49	57.78
MPNet-multilingual	67.91	38.40	80.81	53.80	35.34	80.73	**31.57**	54.71
OpenAI Ada Similarity	70.44	37.52	76.86	49.02	18.36	78.60	26.94	49.52
SGPT-125M-nli	61.46	30.95	71.78	47.56	20.90	74.71	30.26	45.97
SGPT-5.8B-nli	70.14	36.98	77.03	52.33	32.34	80.53	30.38	53.74
SGPT-125M-msmarco	60.72	35.79	75.23	50.58	37.04	73.41	28.90	51.23
SGPT-1.3B-msmarco	66.52	39.92	79.58	54.00	44.49	75.74	25.44	56.11
SGPT-2.7B-msmarco	67.13	39.83	80.65	54.67	46.54	76.83	27.87	57.12
SGPT-5.8B-msmarco	68.13	40.35	82.00	56.56	**50.25**	78.10	24.75	58.81
SGPT-BLOOM-7.1B-msmarco	66.19	38.93	81.90	55.65	48.21	77.74	24.99	57.44
GTR-Base	65.25	38.63	83.85	54.23	44.67	77.07	29.67	56.19
GTR-Large	67.14	41.60	85.33	55.36	47.42	78.19	29.50	58.28
GTR-XL	67.11	41.51	**86.13**	55.96	47.96	77.80	30.21	58.42
GTR-XXL	67.41	42.42	86.12	56.65	48.48	78.38	30.64	58.97
ST5-Base	69.81	40.21	85.17	53.09	33.63	81.14	31.39	55.27
ST5-Large	72.31	41.65	84.97	54.00	36.71	81.83	29.64	57.06
ST5-XL	72.84	42.34	86.06	54.71	38.47	81.66	29.91	57.87
ST5-XXL	**73.42**	**43.71**	85.06	56.43	42.24	**82.63**	30.08	**59.51**

Table 1: Average of the main metric (see Section 3.2) per task per model on MTEB English subsets.

TRAIN AND TEST

Results

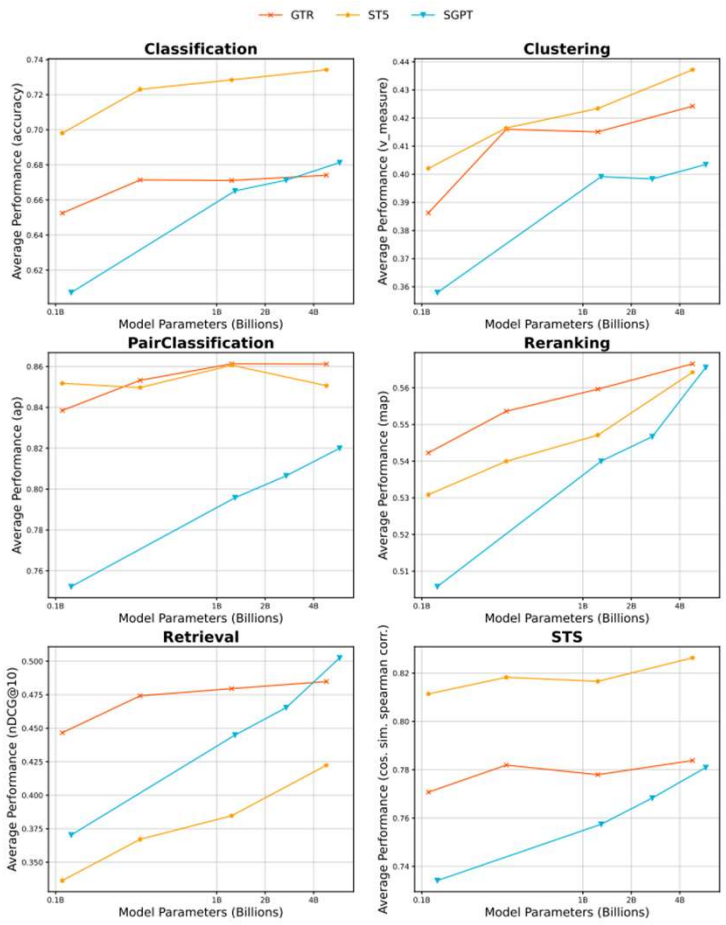
4.2 Analysis

성능은 모델의 크기와 강하게 연관되어 있다!

Performance strongly correlates with model size

하지만 모델의 크기가 크다고 모두 좋은 것은 아니다

Q. 모두 Supervised method 기반 모델인데 왜 SGPT는 사이즈가 작을 때는 다른 두 모델에 비해 성능이 확연히 낮고 커질수록 빠르게 따라잡을까?



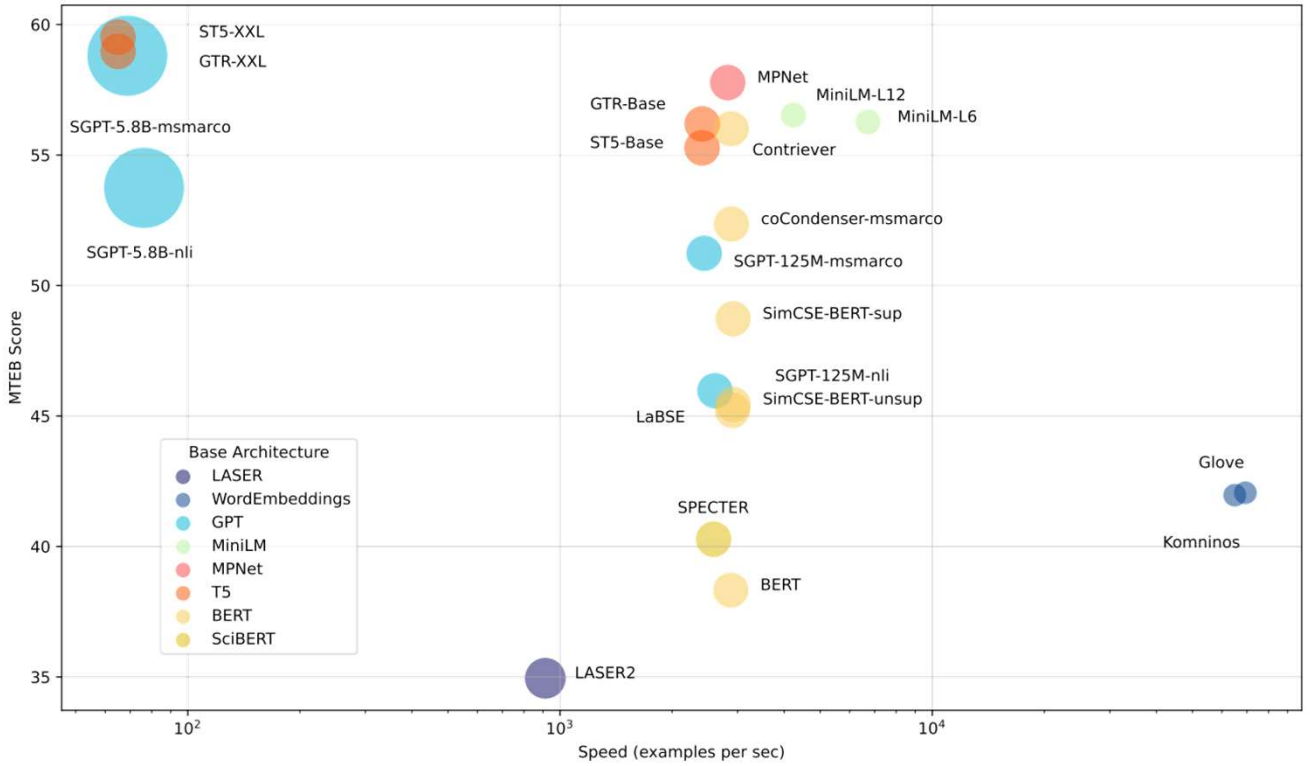
TRAIN AND TEST

Results

4.3 Efficiency

Latency-Performance Trade-off

결국 성능과 속도를 맞바꿔라!



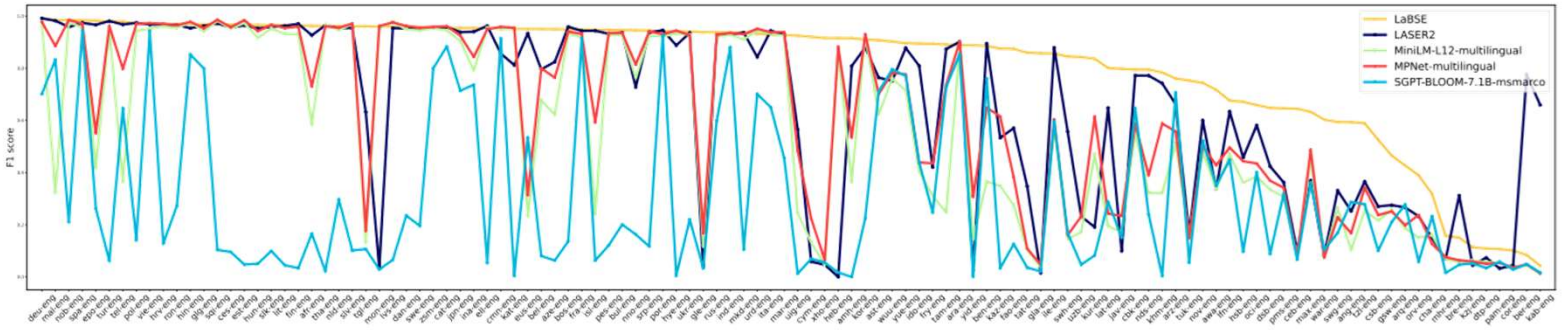
TRAIN AND TEST

Results

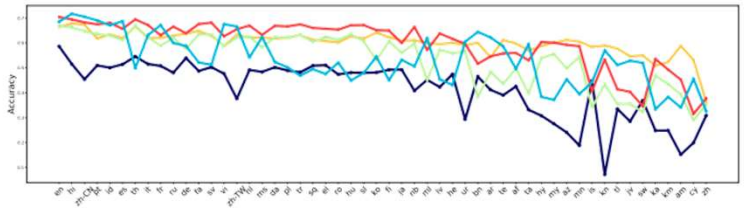
4.4 Multilinguality

10 Multilingual Datasets

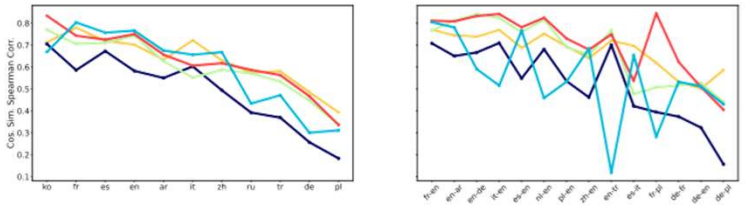
Bitext Mining, Classification, STS



(a) Bitext Mining on Tatoeba



(b) Multilingual Classification



(c) Multi- and Crosslingual STS

TRAIN AND TEST

Conclusion

5. Conclusion

각 Task에서 Model들마다 Performance가 다르다. 하지만 왕중왕은 없었다.

Model Performance on different tasks vary strongly with no model claiming state-of-the-art on all tasks

TRAIN AND TEST

Conclusion

5. Conclusion

Rank	Model	Model Size (Million Parameters)	Memory Usage (GB, fp32)	Embedding Dimensions	Max Tokens	Average (56 datasets)	Classification Average (12 datasets)	Clustering Average (11 datasets)	PairClassification Average (3 datasets)	Reranking Average (4 datasets)	Retrieval Average (15 datasets)	STS Average (10 datasets)	Summarization Average (1 datasets)
1	voyage-large-2-instruct			1024	16000	**68.28**	81.49	53.35	89.24	60.09	58.28	84.58	30.84
2	SFR-Embedding-Mistral	7111	26.49	4096	32768	**67.56**	78.33	51.67	88.54	60.64	59	85.05	31.16
3	gte-Qwen1.5-7B-instruct					**67.34**	79.6	55.83	87.38	60.13			

TRAIN AND TEST

Limitations

6. Limitations of MTEB

1. Long Document Datasets

S2S와 P2P, S2P는 지원 가능하지만 매우 긴 Document는 아직 지원이 힘듬

2. Task Imbalance

Summarization의 경우, 하나의 dataset으로 이루어져 있음
반면, 다른 작업들은 다양한 양의 데이터셋으로 구성되어 있음
MTEB의 Average Score이 많은 데이터셋을 가진 Task에 편향되어 있음

3. Multilinguality

Retrieval과 Clustering은 오로지 English만 지원. 이에 따라 적절히 평가되지 못하는 모델 발생
Code Dataset도 지원하지 않음

4. Additional Modalities

Text Embedding은 Image content와 같은 다른 Modality의 입력으로 사용될 수 있음
이와 같은 활용에 대한 광범위한 벤치마크도 필요함

TRAIN AND TEST

TRAIN AND TEST

References

MTEB: Massive Text Embedding Benchmark , Niklas Muennighof et. al, Mar 19, 2023
MTEB: Massive Text Embedding Benchmark | Video Summary, Niklas Muennighoff, Youtube
Evaluating Embeddings with MTEB Massive text embeddings benchmark - Nils Reimers, Cohere, Youtube
Mechanism of Benchmarking, International Journal of Business Performance Management, Sep, 2016
The importance of Benchmarking, Scott Lenetm Forbes, Dec 12, 2018
Challenges and Opportunities in NLP Benchmarking, Sebastian Ruder, Ruder.io, Aug 23, 2021
Benchmark in Natural Language Processing (NLP), Teerapong Aramruang, Mahidol University
Resources and Benchmarks for NLP, Nico Hahn, https://slds-lmu.github.io/seminar_nlp_ss20/resources-and-benchmarks-for-nlp.html