MTEB：大规模文本嵌入基准测试

摘要

文本嵌入通常在单个任务的小数据集上进行评估，而不涉及它们可能用于其他任务的情况。目前尚不清楚最先进的语义文本相似性（STS）嵌入是否也能同样适用于聚类或重排等其他任务。这使得该领域的进展难以跟踪，因为各种模型不断被提出，但没有得到适当的评价。为了解决这个问题，我们引入了大规模文本嵌入基准测试（MTEB）。MTEB 涵盖了8种嵌入任务、共58个数据集和112种语言。通过在 MTEB 上对33种模型进行基准测试，我们建立了迄今为止最全面的文本嵌入基准测试。我们发现没有任何一种特定的文本嵌入方法在所有任务中都占主导地位。这意味着该领域尚未就通用文本嵌入方法达成共识，并将其扩展到足以在所有嵌入任务上实现最佳结果的程度。MTEB 提供开源代码和公共排行榜，网址为https://github.com/embeddings-benchmark/mteb。

1. 简介

自然语言嵌入为各种用例提供了动力，从聚类和主题表示（Aggarwal 和 Zhai，2012；Angelov，2020）到搜索系统和文本挖掘（Huang 等人，2020；Zhu 等人，2021；Nayak，2019），再到下游模型的特征表示（Saharia 等人，2022；Borgeaud 等人，2022）。由于这些应用通常需要指数级更多的计算量（Reimers 和 Gurevych，2019 年），因此使用生成语言模型或跨编码器在这些应用中通常是不可行的。

然而，当前文本嵌入模型的评估体系很少涵盖这些方面。

它们可能的应用场景。例如，Sim-CSE (Gao等人，2021年b) 或者SBERT(Reimers和Gurevych，2019) 只对STS和分类任务进行评估，而没有对嵌入模型在搜索或聚类任务中的可转移性留下悬而未决的问题。众所周知，STS与其他现实世界用例的相关性很低（Neelakantan等人，2022；Wang等人，2021）。此外，在许多任务上评估嵌入方法需要实现多个评估管道。预处理或超参数等实施细节可能会对结果产生影响，这使得不清楚性能改进是否只是来自有利的评估管道。这导致了这些模型在工业中“盲目”地应用于新的用例，或者需要不断的工作来重新评估它们在不同的任务上的表现。

大规模文本嵌入基准 (MTEB) 的目标是明确模型在各种嵌入任务上的表现，从而作为发现适用于多种任务的通用文本嵌入的门户。MTEB 包含来自 8 种嵌入任务的 58 个数据集：双语挖掘、分类、聚类、配对分类、重排序、检索、STS 和摘要。MTEB 软件已开源，只需添加少于 10 行代码即可评估任何嵌入模型。Hugging Face Hub 上提供数据集和 MTEB 领先榜。

我们在 MTEB 上评估了超过 30 模型，同时进行额外的速度和内存基准测试，以提供对文本嵌入模型状态的整体视图。我们涵盖了开源可用的模型以及通过 API 可用的模型，例如 OpenAI 嵌入式端点。我们发现没有单一的最佳解决方案，不同的模型在不同任务上表现最佳。

我们比较了各种任务。我们的基准研究揭示了单个模型的优缺点，例如SimCSE（Gao等人，2021年）在聚类和检索方面的性能较低，尽管它在STS方面表现良好。我们希望这项工作能够使选择正确的嵌入模型变得更容易，并简化未来的嵌入研究。

2 相关工作

2.1 基准测试

基准测试，如 (Super)GLUE（Wang et al.，2018年和2019年（Big-Shot）或大型样本集(Big-BENCH Srivastava等人，2022)，以及评估框架（Gao等人，2021a）在推动NLP进展方面发挥着关键作用。每年发布的SemEval数据集（Agirre等人，2012、2013、2014、2015、2016）通常用作文本嵌入的基准测试。SemEval数据集对应于语义文本相似性（STS）任务，要求模型对几何上接近的嵌入进行相似句子嵌入。由于单个SemEval数据集表达能力有限，SentEval (Conneau和Kiela，2018)聚合了多个STS数据集。SentEval专注于在嵌入的基础上微调分类器。它缺乏检索或聚类等任务，其中嵌入直接比较而无需额外的分类器。此外，该工具包是在2018年提出的，因此不提供对最近趋势的支持，例如来自transformer的文本嵌入（Reimers和Gurevych，2019）。由于STS基准测试的不足，提出了USEB（Wang等人，2021），主要由重排序任务组成。因此，它不

涵盖检索或分类等任务。与此同时，最近发布的BEIR基准（Thakur等人，2021）已成为零样本信息检索中嵌入评估的标准。

MTEB 将来自不同嵌入任务的数据集统一到一个通用、可访问的评估框架中。MTEB 包括 SemEval 数据集 (STS11– STS22) 和 BEIR，以及来自各种任务的各种其他数据集，以提供文本嵌入模型的整体性能审查。

2.2 嵌入模型

像Glove（Pennington等人，2014年）这样的文本嵌入模型缺乏上下文意识，因此通常被称为词嵌入模型。 它们由一个层组成，该层将每个输入单词映射到向量，随后是平均层以提供与输入长度无关的最终嵌入。

Transformer（Vaswani等人，2017）通过自注意力向语言模型中注入上下文意识，并成为大多数最新嵌入式模型的基础。BERT（Devlin等人，2018）使用Transformer架构并执行大规模自我监督预训练。由此产生的模型可以直接用于生成文本嵌入，方法与Glove类似。SBERT（Reimers和Gurevych，2019），基于InferSent（Conneau等人，2017），表明对Transformer进行额外微调以获得竞争性的嵌入性能是有益的。最近微调的嵌入模型大多使用对比损失目标在正负文本对上进行有标签微调（Gao等人，2021年；王等人，2021年；Ni等人，2021年；Muennighoff，2022）。由于可用的预训练Transformer种类繁多（Wolf等人，2020），因此至少同样存在着各种潜在的文本嵌入模型可供探索。这导致了混淆，不知道哪个模型为从业者提供了最佳的嵌入用例性能。

我们在 MTEB 上对词嵌入和转换器模型进行基准测试，以量化通常较慢的上下文感知模型所提供的增益。

3. MTEB 指数

3.1 理想化要求

MTEB 是基于一组愿望构建的：

(a) 多样性：MTEB 的目标是提供对嵌入模型在各种用例中的可用性的理解。该基准测试包含 8 个不同的任务，每个任务最多有 15 个数据集。MTEB 中的 58 个数据集中有 10 个是多语言的，涵盖 112 种不同的语言。包括句子级别和段落级别的数据集以对比短文本和长文本上的性能。

(b) 简单性：MTEB 提供了一个简单的 API，可以轻松地插入任何模型，给定一个文本列表，它能够为每个列表项生成具有相同形状的向量。这使得可以对多样化的模型进行基准测试。

(c) 扩展性：通过指定任务和数据上传到 Hugging Face 数据集名称的一个文件，可以在 MTEB 中对现有任务的新数据集进行基准测试（Lhoest等人，2021）。新的任务需要实现加载数据的任务接口以及用于基准测试的评估器。我们欢迎社区通过拉取请求提交数据集、任务或指标贡献，以继续开发 MTEB。

(d) 可重复性：通过数据集和软件级别的版本控制，我们旨在使在MTEB中重现结果变得容易。本论文中提供的所有结果的JSON文件与MTEB基准测试包一起提供。

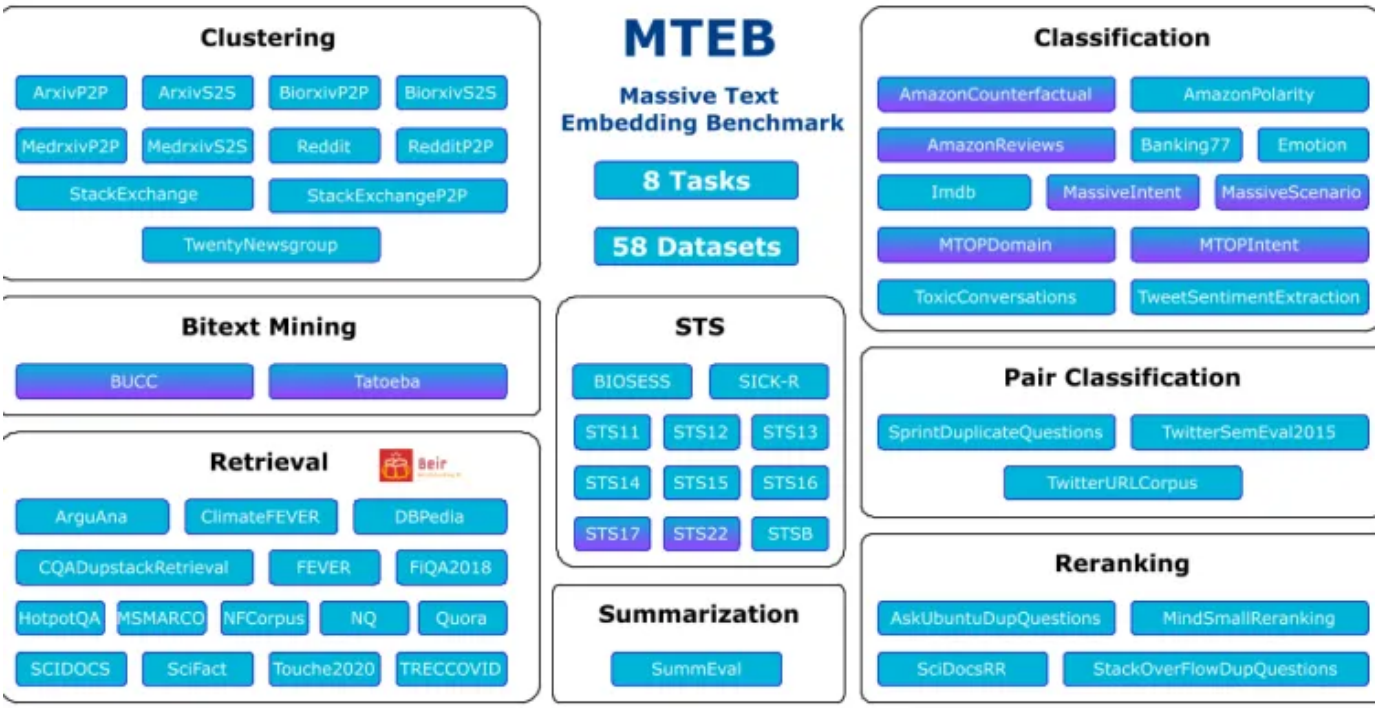


Figure 1: An overview of tasks and datasets in MTEB. Multilingual datasets are marked with a purple shade.

3.2 任务与评价

图 1 展示了 MTEB 中可用的任务和数据集。数据集统计信息可以在表 2 找到。该基准测试包含以下 8 种任务类型：

双语语料库挖掘：输入是来自两种不同语言的两组句子。对于第一组中的每个句子，需要在第二组中找到最佳匹配。匹配通常为翻译。提供的模型用于嵌入每句话，并通过余弦相似度查找最近的一对。F1 作为双语语料库挖掘的主要指标。还计算了准确率、精确率和召回率。

分类：通过提供模型，训练集和测试集嵌入。使用训练集嵌入来训练逻辑回归分类器，最大迭代次数为100，该分类器在测试集上进行评分。主要指标是准确率、平均精度和f1值。

聚类：给定一组句子或段落，目标是将它们分组为有意义的群集。在嵌入文本上训练了一个小批量k均值模型（Pedregosa等人，2011年），批大小为32，k等于不同标签的数量。使用v量表（Rosenberg和Hirschberg，2007）对模型进行评分。V量表不依赖于聚类标签，因此标签的排列不会影响分数。

配对分类：输入两段文本，需要分配一个标签。 标签通常是二进制变量，表示重复或近义词对。 这两个文本被嵌入并使用各种度量（余弦相似度、点积、欧几里得距离、曼哈顿距离）计算它们之间的距离。 使用最佳二元阈值精度、平均精度、F1、精确率和召回率进行计算。 基于余弦相似性的平均精度得分是主要指标。

重排序：输入是一组查询和一组相关或不相关的参考文本。目的是根据它们与查询的相关性对结果进行排名。模型用于嵌入参考文献，然后使用余弦相似度将其与查询进行比较。为每个查询生成一个排名，并对其所有查询取平均值。指标包括均值MRR@k和MAP，其中后者是主要指标。

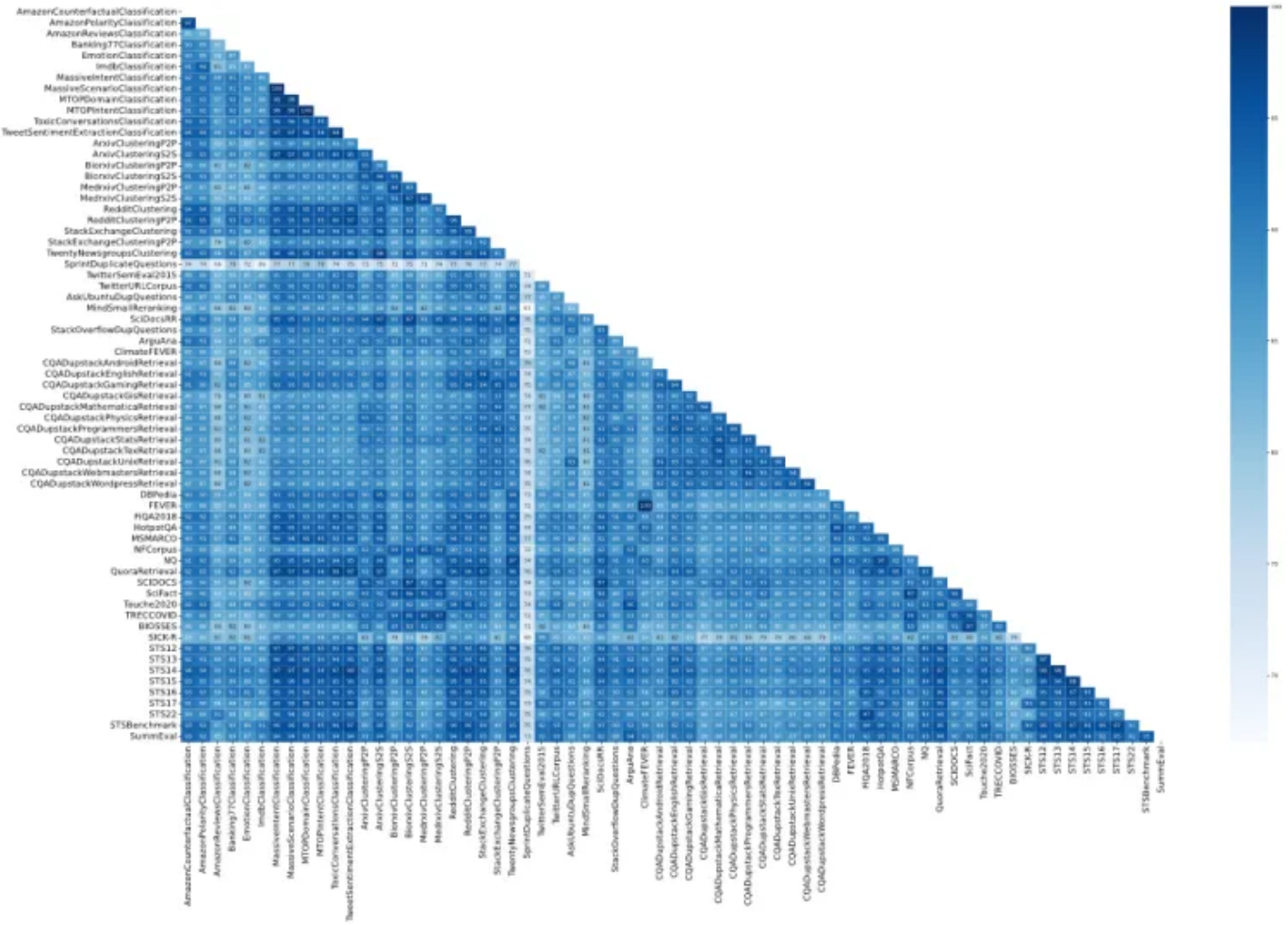


Figure 2: Similarity of MTEB datasets. We use the best model on MTEB STS (ST5-XXL, see Table 1) to embed 100 samples for each dataset. Cosine similarities between the averaged embeddings are computed and visualized.

检索：每个数据集都包含一个语料库、查询和每个查询到语料库中相关文档的映射。目标是找到这些相关的文档。提供的模型用于嵌入所有查询和语料库中的所有文档，并使用余弦相似度计算相似性分数。根据分数对每个查询的语料库文档进行排名后，对于几个值k，可以计算nDCG@k、MRR@k、MAP@k、precision@k和recall@k。nDCG@10作为主要指标。MTEB重用BEIR（Thakur等人，2021）的数据集和评估。

语义文本相似度（STS）：给定一对句子，目的是确定它们之间的相似性。标签是连续得分，数值越高表示越相似的句子。提供的模型用于嵌入句子，并使用各种距离度量计算其相似性。通过皮尔逊相关系数和斯皮尔曼等级相关系数与地面真实相似性进行比较。基于余弦相似性的斯皮尔曼等级相关系数用作主要指标（Reimers等人，2016）。

总结：提供了由人类编写的和机器生成的摘要。目的是对机器生成的摘要进行评分。首先，使用提供的模型对所有摘要进行嵌入。对于每个机器生成的摘要嵌入，计算其与所有人类生成的摘要嵌入之间的距离。保留最近的距离（例如，最高的余弦相似度），并将其用作单个机器生成的摘要的分数。计算与机器生成的摘要的地面真实评估之间的人类评估的相关性（Pearson相关性和Spearman相关性）。如STS所示，基于余弦相似性的Spearman相关性作为主要指标（Reimers等人，2016年）。

3.3 数据集

为了进一步丰富MTEB，我们使用了长度各异的数据集。所有数据集都分为三类：

句子对句子（S2S）：将一个句子与另一个句子进行比较。S2S 的一个例子是在 MTEB 中的所有当前STS任务，其中两个句子之间的相似性得到评估。

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.

段对段（P2P）：一个段落与另一个段落进行比较。MTEB 不限制输入长度，如果需要的话，让模型自行截断。将几个聚类任务构建成S2S 和 P2P 任务。前者只比较标题，而后者包括标题和内容。例如，在ArxivClustering

中，摘要在P2P设置下会连接到标题。

句子到段落（S2P）： 在S2P设置中，混合了几个检索数据集。这里的查询是一个单独的句子，而文档是由多个句子组成的长段落。

图2显示了56个MTEB数据集之间的相似性。其中一些数据集依赖于相同的语料库，如 ClimateFEVER 和 FEVER，导致得分为 1。在 CQADupstack 变体和 STS 数据集之间可以看到类似的子数据集集群。来自同一数据集的 S2S 和 P2P 变体往往也很相似。科学数据集，如 SciDocsRR、SciFact、ArxivClustering，在不同的任务（在这种情况下为排名、检索和聚类）中表现出高度的相似性。

4. 结果

4.1 模型

我们在所有数据集的测试拆分上进行评估，除了 MSMARCO 数据集，其开发拆分使用了 Thakur 等人(2021)的方法。我们基准测试模型，声称在各种嵌入中取得了最先进的结果：

GTR ST5 —SGPT

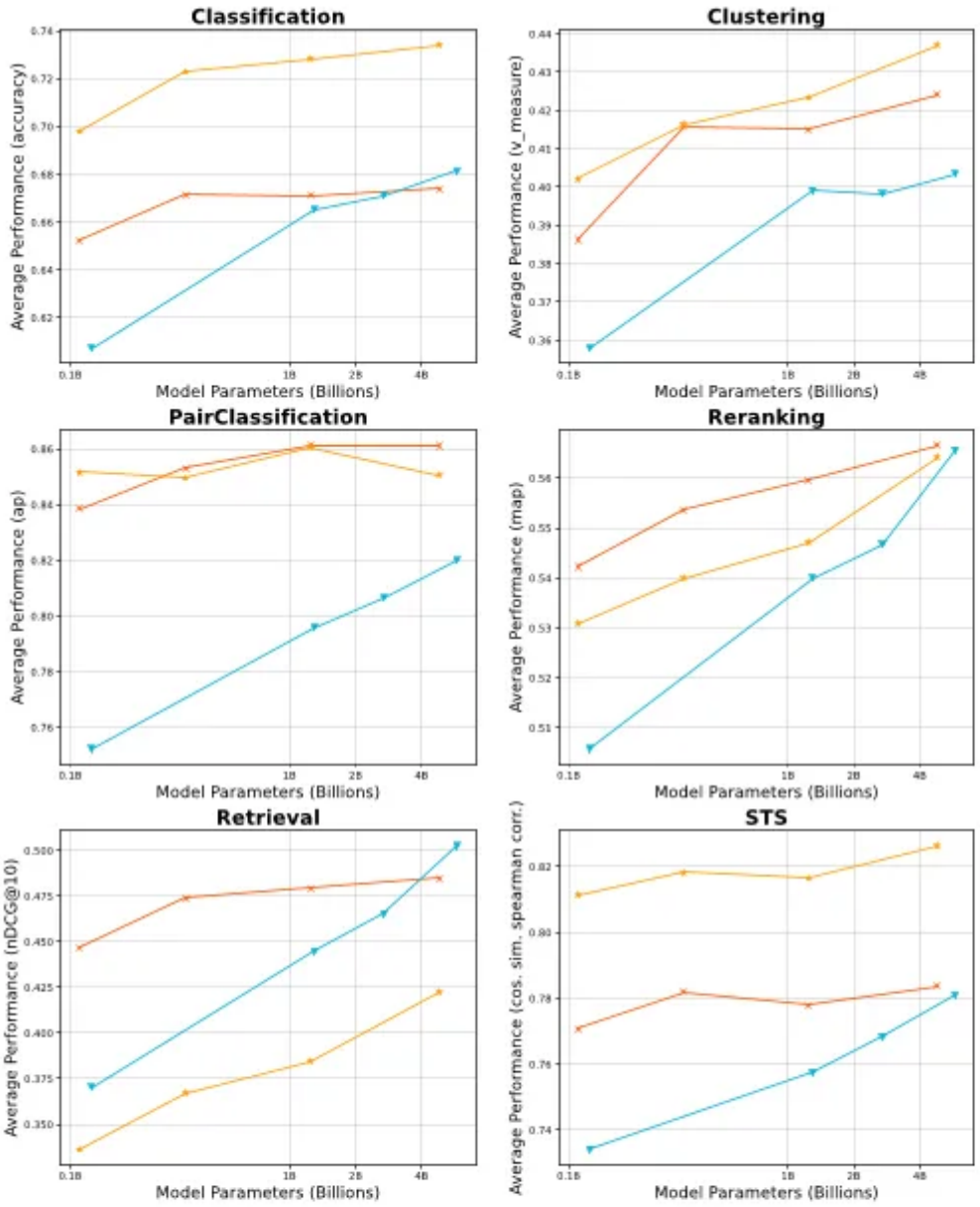


Figure 3: MTEB performance scales with model size. The smallest SGPT variant underperforms similar-sized GTR and ST5 variants. This may be due to the bias-only fine-tuning SGPT employs, which catches up with full fine-tuning only as model size and thus the number of bias parameters increases (Muennighoff, 2022).

(a) Transfo

denser (Gao a
ard et al., 202
SimCSE-BER
on the pre-tra
2018). coCon
supervised sta
for a total of t
BERT to perfo
allel data to pr
model. SPECT
the pre-trained
ant instead and
(Ni et al., 202
based on the e
fel et al., 2020)
datasets. After
ST5 does contr
2021a; Gao et
STS tasks. M
MARCO and
and MiniLM c
models (Reime
trained MPNet
(Wang et al., 2
to target any er

(b) Transfo

我们使用基于奖励的任务来训练模型，以提高其在生成句子（Vaswani 等人，2017 年）中的代表性。我们将模型分为自监督方法和监督方法。

自监督方法(a) 基于Transformer的 BERT（Devlin 等人，2018 年）使用自监督掩码和句子预测任务进行训练。通过在序列长度上取平均值（平均池化），模型可以直接用于生成文本嵌入。SimCSE–Unsup (Gao et al.,

2021b) 使用 BERT 作为基础，并进行额外的自监督训练。(b) 非Transformer：Komninos 和 Manandhar (2016) 和 GloVe (Pennington et al.，2014) 是两种直接将单词映射到向量的词嵌入模型。因此，它们的嵌入缺乏上下文意识，但提供了显着的速度提升。

监督方法 原始的 Transformer 模型（Vaswani 等人，2017 年）由编码器和解码器网络组成。随后的 Transformer 模型通常只训练像 BERT（Devlin 等人，

(2018 年) 或者像 GPT (Radford 等人，2019 年) 这样的解码器。

(a) 基于预训练模型 BERT (Devlin et al., 2018) 的方法包括同构解码器（Gao 和 Callan，2021 年）、检索器（Izacard 等人，2021 年）、LaBSE（Feng 等人，2020 年）和 SimCSE-BERT-sup（Gao 等人，2021 年）。coCondenser 和检索器在监督微调之前添加了一个自监督阶段，总共三个训练阶段。LaBSE 使用 BERT 在并行数据上进行额外的预训练以产生具有竞争力的双语挖掘模型。SPECTER（Cohan 等人，2020 年）依赖于预训练的 SciBERT 变体（Beltagy 等人，2019 年），并在引用图上进行微调。GTR（Ni 等人，2021 年）和 ST5（Ni 等人，2021 年）基于 T5 模型的编码部分（Raffel 等人，2020 年），仅在微调数据集上有所不同。ST5 在额外的自监督训练后对 NLI 进行对比度微调（Ni 等人，2021 年；Gao 等人，2021 年），旨在针对STS任务。与此同时，GTR 在 MS MARCO 上进行微调，并专注于检索任务。MPNet 和 MiniLM 分别对应于使用多样化的数据针对任何嵌入用例进行了微调的预训练 MPNet（Song 等人，2020 年）和 MiniLM（Wang 等人，2020 年）的嵌入模型（Reimers 和 Gurevych，2019 年）。

(b) 使用带有加权平均池化的SGPT双编码器（Muennighoff，2022）对<0.1%的预训练参数进行对比度微调。与ST5和GTR类似，SGPT-nli模型旨在执行STS，而SGPT-msmarco模型用于检索。SGPT-msmarco模型使用不同的特殊标记嵌入查询和文档以帮助模型区分其作用。对于非检索任务，我们使用其查询表示形式。我们在公开可用的基于GPT-NeoX (Andonian等人，2021)，GPT-J (Wang和Komatsuzaki，2021)和BLOOM (Scao等人，2022)的SGPT模型上进行了基准测试。或者，cpt-text (Nee-lakantan等人，2022)通过使用最后令牌池化方法使预先训练的GPT解码器经过两个阶段的过程来提供来自解码器的嵌入。我们通过OpenAI Embeddings API4对其模型进行了基准测试。

（c） 非转换器激光（赫弗南等人，2022年）是我们基准测试中唯一一种基于长短期记忆网络的上下文感知非转换器模型。

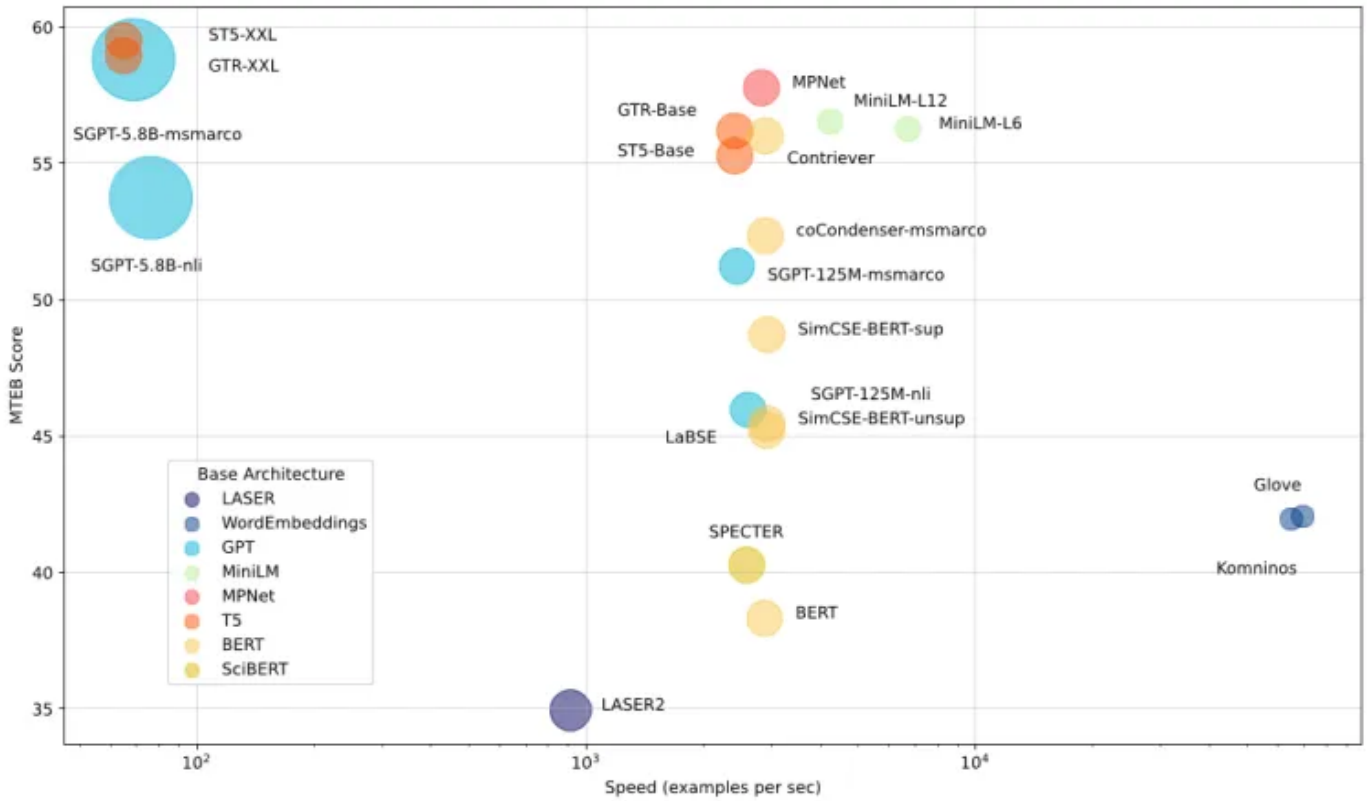


Figure 4: Performance, speed, and size of produced embeddings (size of the circles) of different embedding models. Embedding sizes range from 1.2 kB (Glove / Komninos) to 16.4 kB (SGPT-5.8B) per example. Speed was benchmarked on STS15 using 1x Nvidia A100 80GB with CUDA 11.6.

(Hochreiter 和 Schmidhuber，1997 年）代替。与 LaBSE 类似，该模型在并行数据上进行训练，并专注于双语语料库挖掘应用。

4.2 分析

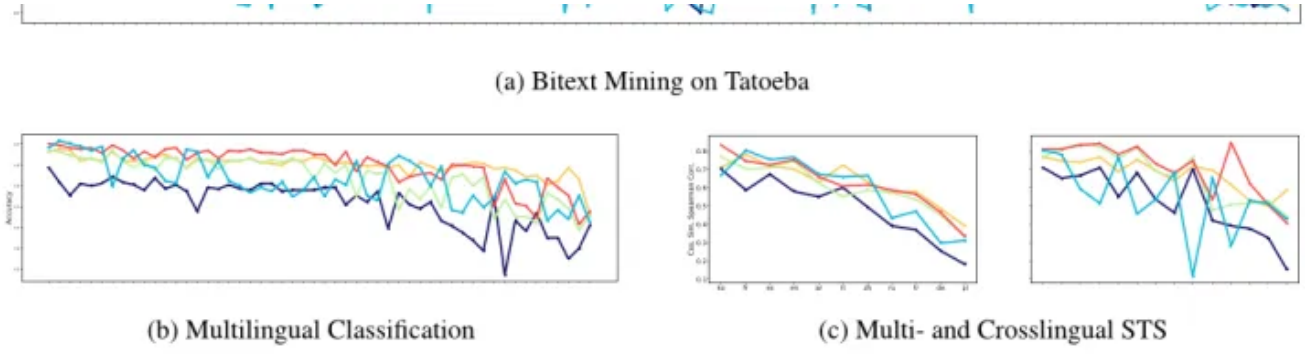
根据表1的结果，我们观察到在任务之间存在相当大的变异性。没有模型声称在所有七项英语任务中都达到最先进的水平。附录中的每个数据集的结果甚至更具多样性。此外，在自监督和监督方法之间仍然存在着很大的差距。自监督大型语言模型已经在许多自然语言生成任务中缩小了这一差距（Chowdhery等人，2022）。然而，它们似乎仍需要监督微调以获得有竞争力的嵌入性能。

我们发现性能与模型大小有很强的相关性，见图3。 大多数 MTEB 任务由数十亿参数的模型主导。 然而，正如我们在第 4.3 节中所讨论的，这些模型需要付出高昂的代价。

ST5 模型在大多数数据集上主导分类任务，正如附录中的完整结果所示。ST5-XXL 的平均性能最高，比最好的非 ST5 模型，OpenAI Ada 相似性。

聚类 尽管 MPNet嵌入模型 的大小几乎只有ST5-XXL的50倍，但在聚类方面却与之相当。这可能是因为MPNet（以及MiniLM）在训练过程中微调了大量数据集。聚类需要大量的嵌入向量之间保持一致的距离。只在一个数据集NLI上进行微调的模型，如SimCSE-sup或SGPT-nli，在遇到微调期间未见过的主题时可能会产生不连贯的嵌入。相关性地，我们发现SGPT-msmarco的查询嵌入和Ada搜索端点与SGPT-nli和Ada相似度端点具有竞争

力。我们查阅了公共排行榜5上的Ada搜索结果。这可能是由于MSMARCO数据集比NLI大得多。因此，虽然OpenAI文档建议我们在聚类用例中使用相似度嵌入6，但检索查询嵌入在某些情况下可能是更好的选择。



(a) Bitext Mining on Tatoeba

(b) Multilingual Classification

(c) Multi- and Crosslingual STS

Figure 5: MTEB multilingual performance. Bitext mining is dominated by LaBSE, while classification and STS results are mixed. SGPT-BLOOM-7B1-msmarco tends to perform well on the languages BLOOM has been pre-trained on, such as Chinese, French and Portuguese.

对齐分类 GTR–XL 和 GTR–XXL 的性能最强。对齐分类在框架上最接近 STS，但模型在这两项任务上的排名显著不同。这强调了在各种任务上进行基准测试的重要性，以避免盲目地重复使用模型来处理不同的任务。

重排模型 MPNet 和 MiniLM 在重排任务上表现良好。在 SciDocsRR（Co–han 等，2020 年）中，它们的表现远胜于大型模型，这可能是因为 SciDocsRR 的部分数据包含在其训练数据中。我们的实验规模和预训练模型使控制数据污染变得具有挑战性。因此，在 MTEB 得分中，我们忽略了 MTEB 数据集与模型训练数据集之间的重叠。只要足够多的数据被平均，我们认为这些影响是可以忽略不计的。

Retrieval SGPT–5.8B–msmarco 是 MTEB 中在 BEIR 子集以及整个 BEIR 标准数据集中最好的嵌入模型（Thakur 等，2021 年；Muennighoff，2022 年）。使用 BLOOM 的更大 7.1B SGPT 模型表现明显较差，这可能是因为 BLOOM 的多语言性。针对STS的任务（SimCSE、ST5、SGPT–nli）的模型在检索任务中表现不佳。检索任务的独特之处在于有两种不同的文本类型：查询和文档（“不对称”），而其他任务只有一种文本类型（“对称”）。在 QuoraRetrieval 数据集上，在该数据集已证明其大部分是对称性的。

（Muennighoff，2022），SGPT–5.8B–nli在SGPT–5.8B–smarco中表现更好，见表11。

STS 和摘要检索模型（GTR、SGPT MsMarco）在 STS 上表现不佳，而 ST5–XXL 的性能最高。这突显了该领域分裂为针对检索（非对称）和相似性（对称）用例的不同嵌入模型的趋势（Muennighoff, 2022 年）。

4.3 效率

我们研究了图 4 中模型的延迟性能权衡。该图表允许在模型选择过程中显着消除候选模型，将其减少到三个群集：

速度最快词嵌入模型在性能和速度方面都以 glove 为首，因此在这种情况下选择变得简单。

最大性能 如果延迟比性能不那么重要，图表左侧提供了一组高性能但缓慢的模型。具体取决于手头的任务，GTR–XXL、ST5–XXL 或 SGPT–5.8B 可能是正确的选择，参见第 4.2 节。SGPT–5.8B 的额外缺点是其高维嵌入需要更多的存储空间。

速度与性能 调整后的 MPNet 和 MiniLM 模型处于中间集群，使选择变得容易。

4.4 多语言

MTEB 包含 10 种多语言数据集，用于双语挖掘、分类和STS任务。我们在图 5 中研究了这些方面的性能。表格结果可以在表 12、表 13 和表 14 中找到。

双语挖掘 LaBSE (Feng等人，2020年) 在广泛的语言中表现出强大的性能。与此同时，LASER2在不同的语言上显示出很高的方差。虽然我们基准测试的一些语言有额外的语言特定 LASER2 模型可用，但我们在所有语言上都使用了默认的多语言 LASER2 模型。这是为了提供模型之间的公平一对一比较。然而，在实践中，通过混合 LASER2 的模型变体可以解决其性能的高方差。MP-Net、MiniLM 和 SGPT-BLOOM-7B1-msmarco 在它们没有预先训练过的语言（例如后者的德语）上表现较差。

分类与STS 多语言分类和多语言语义相似性度量方面，多语言MPNet提供了整体上最强的效果。它在几乎所有语言上都优于略快的多语言MiniLM。 这两个模型都在相同语言上进行了训练，因此决策权衡的是性能对速度的影响。 SGPT-BLOOM-7B1-msmarco 在诸如印地语、葡萄牙语、汉语或法语等语言上提供了最先进的性能，这些语言在预训练期间被该模型广泛使用。它还以竞争力的方式处理俄语或日语等语言（Muennighoff等人，2022），这些语言无意中泄露到其预训练数据中（Muennighoff等人，2022）。然而，它并没有比便宜得多的MPNet领先太多。 LASER2始终表现不如其他模型。

5 结论

在这项工作中，我们介绍了大规模文本嵌入基准测试集（Massive Text Embedding Benchmark, MTEB）。它由多达 15 个数据集的任务组成，涵盖 112 种语言，旨在提供可靠的嵌入性能估计。通过公开 MTEB 和排行榜，我们为推动可用文本嵌入的最先进技术提供了基础。

为了介绍MTEB，我们进行了迄今为止最全面的文本嵌入基准测试。通过在超过30种不同的模型上进行近5000次实验，我们为未来的研究建立了坚实的基础。

我们发现模型在不同任务上的表现存在很大差异，没有一个模型能够在所有任务上达到最先进的水平。我们对缩放行为、模型效率和多语言能力的研究揭示了模型的各种复杂性，这应该有助于简化未来研究或工业应用中的嵌入式文本决策过程。

我们欢迎 任务、数据集或度量 的贡献，以补充MTEB代码库7，以及通过我们的自动提交格式8添加到排行榜中。