MTEB: Massive Text Embedding Benchmark

Niklas Muennighoff¹, Nouamane Tazi¹, Loïc Magne¹, Nils Reimers²*
¹Hugging Face  ²cohere.ai
¹firstname@huggingface.co  ²info@nils-reimers.de

Abstract

Text embeddings are commonly evaluated on a small set of datasets from a single task not covering their possible applications to other tasks. It is unclear whether state-of-the-art embeddings on semantic textual similarity (STS) can be equally well applied to other tasks like clustering or reranking. To solve this problem, we introduce the Massive Text Embedding Benchmark (MTEB). MTEB spans 8 embedding tasks covering a total of 58 datasets and 112 languages. Through the benchmarking of 33 models on MTEB, we establish the most comprehensive benchmark of text embeddings to date. We find that no particular text embedding method dominates across all tasks. This suggests that the field has yet to converge on a universal text embedding method and scale it up sufficiently to provide state-of-the-art results on all embedding tasks. MTEB comes with open-source code and a public leaderboard at https://github.com/embbeddings-benchmark/mteb.

1 Introduction

Natural language embeddings power a variety of use cases from clustering and topic representation (Aggarwal and Zhai, 2012; Angelov, 2020) to search systems and text mining (Huang et al., 2020; Zhu et al., 2021; Nayak, 2019) to feature representations for downstream models (Saharia et al., 2022; Borgeaud et al., 2022). Using generative language models or cross-encoders for these applications is often intractable, as they may require exponentially more computations (Reimers and Gurevych, 2019).

However, the evaluation regime of current text embedding models rarely covers the breadth of their possible use cases. For example, SimCSE (Gao et al., 2021b) or SBERT (Reimers and Gurevych, 2019) solely evaluate on STS and classification tasks, leaving open questions about the transferability of the embedding models to search or clustering tasks. STS is known to poorly correlate with other real-world use cases (Neelakantan et al., 2022; Wang et al., 2021). 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. This leads to the “blind” application of these models to new use cases in industry or requires incremental work to reevaluate them on different tasks.

The Massive Text Embedding Benchmark (MTEB) aims to provide clarity on how models perform on a variety of embedding tasks and thus serves as the gateway to finding universal text embeddings applicable to a variety of tasks. MTEB consists of 58 datasets covering 112 languages from 8 embedding tasks: Bitext mining, classification, clustering, pair classification, reranking, retrieval, STS and summarization. MTEB software is available open-source¹ enabling evaluation of any embedding model by adding less than 10 lines of code. Datasets and the MTEB leaderboard are available on the Hugging Face Hub².

We evaluate over 30 models on MTEB with additional speed and memory benchmarking to provide a holistic view of the state of text embedding models. We cover both models available open-source as well as models accessible via APIs, such as the OpenAI Embeddings endpoint. We find there to be no single best solution, with different models dominating different tasks. Our benchmarking sheds light on the weaknesses and strengths of individual

¹https://github.com/embbeddings-benchmark/mteb
²https://huggingface.co/spaces/mteb/leaderboard
models, such as SimCSE’s (Gao et al., 2021b) low performance on clustering and retrieval despite its strong performance on STS. We hope our work makes selecting the right embedding model easier and simplifies future embedding research.

2 Related Work

2.1 Benchmarks

Benchmarks, such as (Super)GLUE (Wang et al., 2018, 2019) or Big-BENCH (Srivastava et al., 2022), and evaluation frameworks (Gao et al., 2021a) play a key role in driving NLP progress. Yearly released SemEval datasets (Agirre et al., 2012, 2013, 2014, 2015, 2016) are commonly used as the go-to benchmark for text embeddings. SemEval datasets correspond to the task of semantic textual similarity (STS) requiring models to embed similar sentences with geometrically close embeddings. Due to the limited expressivity of a single SemEval dataset, SentEval (Conneau and Kiela, 2018) aggregates multiple STS datasets. SentEval focuses on fine-tuning classifiers on top of embeddings. It lacks tasks like retrieval or clustering, where embeddings are directly compared without additional classifiers. Further, the toolkit was proposed in 2018 and thus does not provide easy support for recent trends like text embeddings from transformers (Reimers and Gurevych, 2019). Due to the insufficiency of STS benchmarking, USEB (Wang et al., 2021) was introduced consisting mostly of reranking tasks. Consequently, it does not cover tasks like retrieval or classification. Meanwhile, the recently released BEIR Benchmark (Thakur et al., 2021) has become the standard for the evaluation of embeddings for zero-shot information retrieval.

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.

2.2 Embedding Models

Text embedding models like Glove (Pennington et al., 2014) lack context awareness and are thus commonly labeled as Word Embedding Models. They consist of a layer mapping each input word to a vector often followed by an averaging layer to provide a final embedding invariant of input length. Transformers (Vaswani et al., 2017) inject context awareness into language models via self-attention and form the foundation of most recent embedding models. BERT (Devlin et al., 2018) uses the transformer architecture and performs large-scale self-supervised pre-training. The resulting model can directly be used to produce text embeddings via an averaging operation alike Glove. Building on InferSent (Conneau et al., 2017), SBERT (Reimers and Gurevych, 2019) demonstrated it to be beneficial to perform additional fine-tuning of the transformer for competitive embedding performance. Most recent fine-tuned embedding models use a contrastive loss objective to perform supervised fine-tuning on positive and negative text pairs (Gao et al., 2021b; Wang et al., 2021; Ni et al., 2021b; Muennighoff, 2022). Due to the large variety of available pre-trained transformers (Wolf et al., 2020), there is an at least equally large variety of potential text embedding models to be explored. This leads to confusion about which model provides practitioners with the best performance for their embedding use case.

We benchmark both word embedding and transformer models on MTEB quantifying gains provided by often much slower context aware models.

3 The MTEB Benchmark

3.1 Desiderata

MTEB is built on a set of desiderata: (a) Diversity: MTEB aims to provide an understanding of the usability of embedding models in various use cases. The benchmark comprises 8 different tasks, with up to 15 datasets each. Of the 58 total datasets in MTEB, 10 are multilingual, covering 112 different languages. Sentence-level and paragraph-level datasets are included to contrast performance on short and long texts. (b) Simplicity: MTEB provides a simple API for plugging in any model that given a list of texts can produce a vector for each list item with a consistent shape. This makes it possible to benchmark a diverse set of models. (c) Extensibility: New datasets for existing tasks can be benchmarked in MTEB via a single file that specifies the task and a Hugging Face dataset name where the data has been uploaded (Lhoest et al., 2021). New tasks require implementing a task interface for loading the data and an evaluator for benchmarking. We welcome dataset, task or metric contributions from the community via pull requests to continue the development of MTEB. (d) Reproducibility: Through versioning at a dataset and software level, we aim to make it easy to repro-
duce results in MTEB. JSON files corresponding to all results available in this paper have been made available together with the MTEB benchmark.

3.2 Tasks and Evaluation

Figure 1 provides an overview of tasks and datasets available in MTEB. Dataset statistics are available in Table 2. The benchmark consists of the following 8 task types:

**Bitext Mining** Inputs are two sets of sentences from two different languages. For each sentence in the first set, the best match in the second set needs to be found. The matches are commonly translations. The provided model is used to embed each sentence and the closest pairs are found via cosine similarity. F1 serves as the main metric for bitext mining. Accuracy, precision and recall are also computed.

**Classification** A train and test set are embedded with the provided model. The train set embeddings are used to train a logistic regression classifier with 100 maximum iterations, which is scored on the test set. The main metric is accuracy with average precision and f1 additionally provided.

**Clustering** Given a set of sentences or paragraphs, the goal is to group them into meaningful clusters. A mini-batch k-means model with batch size 32 and k equal to the number of different labels trained on the embedded texts. The model is scored using v-measure. V-measure does not depend on the cluster label, thus the permutation of labels does not affect the score.

**Pair Classification** A pair of text inputs is provided and a label needs to be assigned. Labels are typically binary variables denoting duplicate or paraphrase pairs. The two texts are embedded and their distance is computed with various metrics (cosine similarity, dot product, euclidean distance, manhattan distance). Using the best binary threshold accuracy, average precision, f1, precision and recall are computed. The average precision score based on cosine similarity is the main metric.

**Reranking** Inputs are a query and a list of relevant and irrelevant reference texts. The aim is to rank the results according to their relevance to the query. The model is used to embed the references which are then compared to the query using cosine similarity. The resulting ranking is scored for each query and averaged across all queries. Metrics are mean MRR@k and MAP with the latter being the main metric.

**Retrieval** Each dataset consists of a corpus, queries and a mapping for each query to relevant documents from the corpus. The aim is to find these relevant documents. The provided model is used
to embed all queries and all corpus documents and similarity scores are computed using cosine similarity. After ranking the corpus documents for each query based on the scores, \( \text{nDCG} @k \), \( \text{MRR} @k \), \( \text{MAP} @k \), precision@k and recall@k are computed for several values of \( k \). \( \text{nDCG} @10 \) serves as the main metric. MTEB reuses datasets and evaluation from BEIR (Thakur et al., 2021).

**Semantic Textual Similarity (STS)** Given a sentence pair the aim is to determine their similarity. Labels are continuous scores with higher numbers indicating more similar sentences. The provided model is used to embed the sentences and their similarity is computed using various distance metrics. Distances are benchmarked with ground truth similarities using Pearson and Spearman correlations. Spearman correlation based on cosine similarity serves as the main metric (Reimers et al., 2016).

**3.3 Datasets**

To further the diversity of MTEB, datasets of varying text lengths are included. All datasets are grouped into three categories:

**Sentence to sentence (S2S)** A sentence is compared with another sentence. An example of S2S are all current STS tasks in MTEB, where the similarity between two sentences is assessed.

**Paragraph to paragraph (P2P)** A paragraph is compared with another paragraph. MTEB imposes no limit on the input length, leaving it up to the
models to truncate if necessary. Several clustering tasks are framed as both S2S and P2P tasks. The former only compare titles, while the latter include both title and content. For ArxivClustering, for example, abstracts are concatenated to the title in the P2P setting.

Sentence to paragraph (S2P) A few retrieval datasets are mixed in a S2P setting. Here a query is a single sentence, while documents are long paragraphs consisting of multiple sentences.

Similarities across 56 MTEB datasets are visualized in Figure 2. Several datasets rely on the same corpora, such as ClimateFEVER and FEVER, resulting in a score of 1. Clusters of similar datasets can be seen among CQADupstack variations and STS datasets. S2S and P2P variations of the same dataset tend to also be similar. Scientific datasets, such as SciDocsRR, SciFact, ArxivClustering, show high similarities among each other even when coming from different tasks (Reranking, Retrieval and Clustering in this case).

4 Results

4.1 Models

We evaluate on the test splits of all datasets except for MSMARCO, where the dev split is used following Thakur et al. (2021). We benchmark models claiming state-of-the-art results on various embedding tasks leading to a high representation of transformers (Vaswani et al., 2017). We group models into self-supervised and supervised methods.
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).

Self-supervised methods (a) Transformer-based BERT (Devlin et al., 2018) is trained using self-supervised mask and sentence prediction tasks. By taking the mean across the sequence length (mean-pooling) the model can directly be used to produce text embeddings. SimCSE-Unsup (Gao et al., 2021b) uses BERT as a foundation and performs additional self-supervised training. (b) Non-transformer: Komninos (Komninos and Manandhar, 2016) and Glove (Pennington et al., 2014) are two word embedding models that directly map words to vectors. Hence, their embeddings lack context awareness, but provide significant speed-ups.

Supervised methods The original transformer model (Vaswani et al., 2017) consists of an encoder and decoder network. Subsequent transformers often train only encoders like BERT (Devlin et al., 2018) or decoders like GPT (Radford et al., 2019).

(a) Transformer encoder methods coCondenser (Gao and Callan, 2021), Contriever (Izacard et al., 2021), LaBSE (Feng et al., 2020) and SimCSE-BERT-sup (Gao et al., 2021b) are based on the pre-trained BERT model (Devlin et al., 2018). coCondenser and Contriever add a self-supervised stage prior to supervised fine-tuning for a total of three training stages. LaBSE uses BERT to perform additional pre-training on parallel data to produce a competitive bitext mining model. SPECTER (Cohan et al., 2020a) relies on the pre-trained SciBERT (Beltagy et al., 2019) variant instead and fine-tunes on citation graphs. GTR (Ni et al., 2021b) and ST5 (Ni et al., 2021a) are based on the encoder part of the T5 model (Raffel et al., 2020) and only differ in their fine-tuning datasets. After additional self-supervised training, ST5 does contrastive fine-tuning on NLI (Ni et al., 2021a; Gao et al., 2021b) being geared towards STS tasks. Meanwhile, GTR fine-tunes on MS-MARCO and focuses on retrieval tasks. MPNet and MiniLM correspond to fine-tuned embedding models (Reimers and Gurevych, 2019) of the pre-trained MPNet (Song et al., 2020) and MiniLM (Wang et al., 2020) models using diverse datasets to target any embedding use case.

(b) Transformer decoder methods SGPT Bi-Encoders (Muennighoff, 2022) perform contrastive fine-tuning of <0.1% of pre-trained parameters using weighted-mean pooling. Similar to ST5 and GTR, SGPT-nli models are geared towards STS, while SGPT-msmarco models towards retrieval. SGPT-msmarco models embed queries and documents for retrieval with different special tokens to help the model distinguish their role. For non-retrieval tasks, we use its query representations. We benchmark publicly available SGPT models based on GPT-NeoX (Andonian et al., 2021), GPT-J (Wang and Komatsuzaki, 2021) and BLOOM (Scao et al., 2022). Alternatively, cpt-text (Nee-lakantan et al., 2022) passes pre-trained GPT decoders through a two-stage process using last token pooling to provide embeddings from decoders. We benchmark their models via the OpenAI Embeddings API⁴.

(c) Non-transformer LASER (Heffernan et al., 2022) is the only context aware non-transformer model we benchmark, relying on an LSTM (Hochreiter and Schmidhuber, 1997) instead. Similar to LaBSE, the model trains on parallel data and focuses on bitext mining applications.

⁴https://beta.openai.com/docs/guides/embeddings
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.

4.2 Analysis

Based on the results in Table 1, we observe that there is considerable variability between tasks. No model claims the state-of-the-art in all seven English tasks. There is even more variability in the results per dataset present in the appendix. Further, there remains a large gap between self-supervised and supervised methods. Self-supervised large language models have been able to close this gap in many natural language generation tasks (Chowdhery et al., 2022). However, they appear to still require supervised fine-tuning for competitive embedding performance.

We find that performance strongly correlates with model size, see Figure 3. A majority of MTEB tasks are dominated by multi-billion parameter models. However, these come at a significant cost as we investigate in Section 4.3.

Classification ST5 models dominate the classification task across most datasets, as can be seen in detail in the full results in the appendix. ST5-XXL has the highest average performance, 3% ahead of the best non-ST5 model, Ada Similarity.

Clustering Despite being almost 50x smaller, the MPNet embedding model is on par with the ST5-XXL state-of-the-art on Clustering. This may be due to the large variety of datasets MPNet (and MiniLM) has been fine-tuned on. Clustering requires coherent distances between a large number of embeddings. Models like SimCSE-sup or SGPT-nli, which are only fine-tuned on a single dataset, NLI, may produce incoherent embeddings when encountering topics unseen during fine-tuning. Relatedly, we find that the query embeddings of SGPT-msmarco and the Ada Search endpoint are competitive with SGPT-nli and the Ada Similarity endpoint, respectively. We refer to the public leaderboard5 for Ada Search results. This could be due to the MSMARCO dataset being significantly larger than NLI. Thus, while the OpenAI docs recommend using the similarity embeddings for clustering use cases6, the retrieval query embeddings may be the better choice in some cases.

Pair Classification GTR-XL and GTR-XXL have the strongest performance. Pair classification is closest to STS in its framing, yet models rank significantly differently on the two tasks. This

---

5https://huggingface.co/spaces/mteb/leaderboard
6https://beta.openai.com/docs/guides/embeddings/similarity-embeddings
MTEB comes with 10 multilingual datasets across bitext mining, classification and STS tasks. We investigate the latency-performance trade-off of models in Figure 4. The graph allows for significant elimination of model candidates in the model selection process. It brings model selection down to three clusters:

**Maximum speed**  Word Embedding models offer maximum speed with Glove taking the lead on both performance and speed, thus making the choice simple in this case.

**Maximum performance**  If latency is less important than performance, the left-hand side of the graph offers a cluster of highly performant, but slow models. Depending on the task at hand, GTR-XXL, ST5-XXL or SGPT-5.8B may be the right choice, see Section 4.2. SGPT-5.8B comes with the additional caveat of its high-dimensional embeddings requiring more storage.

**Speed and performance**  The fine-tuned MPNet and MiniLM models lead the middle cluster making the choice easy.

### 4.4 Multilinguality

MTEB comes with 10 multilingual datasets across bitext mining, classification and STS tasks. We in-
investigate performance on these in Figure 5. Tabular results can be found in Tables 12, 13 and 14.

**Bitext Mining** LaBSE (Feng et al., 2020) performs strongly across a wide array of languages in bitext mining. Meanwhile, LASER2 shows high variance across different languages. While there are additional language-specific LASER2 models available for some of the languages we benchmark, we use the default multilingual LASER2 model for all languages. This is to provide a fair one-to-one comparison of models. In practice, however, the high variance of LASER2’s performance may be resolved by mixing its model variants. MPNet, MiniLM and SGPT-BLOOM-7B1-msmarco perform poorly on languages they have not been pre-trained on, such as German for the latter.

**Classification & STS** On multilingual classification and STS, the multilingual MPNet provides the overall strongest performance. It outperforms the slightly faster multilingual MiniLM on almost all languages. Both models have been trained on the same languages, thus bringing decision-making down to performance vs speed. SGPT-BLOOM-7B1-msmarco provides state-of-the-art performance on languages like Hindi, Portuguese, Chinese or French, which the model has seen extensively during pre-training. It also performs competitively on languages like Russian or Japanese that unintentionally leaked into its pre-training data (Muennighoff et al., 2022). However, it is not much ahead of the much cheaper MPNet. LASER2 performs consistently worse than other models.

5 Conclusion

In this work, we presented the Massive Text Embedding Benchmark (MTEB). Consisting of 8 text embedding tasks with up to 15 datasets each and covering 112 languages, MTEB aims to provide reliable embedding performance estimates. By open-sourcing MTEB alongside a leaderboard, we provide a foundation for further pushing the state-of-the-art of available text embeddings.

To introduce MTEB, we have conducted the most comprehensive benchmarking of text embeddings to date. Through the course of close to 5,000 experiments on over 30 different models, we have set up solid baselines for future research to build on. We found model performance on different tasks to vary strongly with no model claiming state-of-the-art on all tasks. Our studies on scaling behavior, model efficiency and multilinguality revealed various intricacies of models that should ease the decision-making process for future research or industry applications of text embeddings.

We welcome task, dataset or metric contributions to the MTEB codebase7 as well as additions to the leaderboard via our automatic submission format8.

6 Limitations of MTEB

While MTEB aims to be a diverse benchmark to provide holistic performance reviews, the benchmark has its limitations. We list them here:

1. **Long document datasets** MTEB covers multiple text lengths (S2S, P2P, S2P), but very long documents are still missing. The longest datasets in MTEB have a few hundred words, and longer text sizes could be relevant for use cases like retrieval.

2. **Task imbalance** Tasks in MTEB have a different amount of datasets with summarization consisting of only a single dataset. This means MTEB average scores, which are computed over all datasets, are biased towards tasks with many datasets, notably retrieval, classification and clustering. As MTEB grows, we hope to add more datasets to currently underrepresented tasks like summarization or pair classification.

3. **Multilinguality** MTEB contains multilingual classification, STS and bitext mining datasets. However, retrieval and clustering are English-only. SGPT-BLOOM-7B1-msmarco is geared towards multilingual retrieval datasets and due to the lack thereof cannot be comprehensively benchmarked in MTEB. Further, MTEB does not contain any code datasets that could be used to benchmark code models (Neelakantan et al., 2022; Allal et al., 2023). It should be easy to extend MTEB with datasets, such as CodeSearchNet (Husain et al., 2019), TyDI QA (Clark et al., 2020), XOR QA (Asai et al., 2020) or MIRACL (Zhang et al., 2022).

4. **Additional modalities** Text embeddings are commonly used as input features for downstream models, such as in our classification task. This can involve other modalities, notably image content (Carvalho et al., 2018; Tan and Bansal, 2019; Muennighoff, 2020; Nichol et al., 2021; Saharia

7https://github.com/embeddings-benchmark/mteb
8https://huggingface.co/spaces/mteb/leaderboard
We have focused solely on natural language applications and leave extensive benchmarking of text embeddings as inputs for other modalities to future work.

Acknowledgments

This work was granted access to the HPC resources of Institut du développement et des ressources en informatique scientifique (IDRIS) du Centre national de la recherche scientifique (CNRS) under the allocation 2021-A0101012475 made by Grand équipement national de calcul intensif (GENCI). In particular, all the evaluations and data processing ran on the Jean Zay cluster of IDRIS, and we want to thank the IDRIS team for responsive support throughout the project, in particular Rémi Lacroix.

We thank Douwe Kiela, Teven Le Scao and Nandan Thakur for feedback and suggestions.

References


Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Shivanshu Purohit, Tri Songz, Phil Wang, and Samuel Weinbach. 2021. GPT-NeoX: Large scale autoregressive language modeling in pytorch.


A Datasets

Table 2 provides a summary along with statistics of all MTEB tasks. In the following, we give a brief description of each dataset included in MTEB.

A.1 Clustering

ArxivClusteringS2S, ArxivClusteringP2P, BiorxivClusteringS2S, BiorxivClusteringP2P, MedrxivClusteringP2P, MedrxivClusteringS2S

These datasets are custom-made for MTEB using the public APIs from arXiv\(^9\) and bioRxiv/medRxiv\(^10\). For S2S datasets, the input text is simply the title of the paper, while for P2P the input text is the concatenation of the title and the abstract. The cluster labels are generated using categories given to the papers by humans. For bioRxiv and medRxiv this category is unique, but for arXiv multiple categories can be given to a single paper so we only use the first one. For bioRxiv and medRxiv there is only one level of category (e.g. biochemistry, genetics, microbiology, etc.) hence we only perform clustering based on that label. For arXiv there is a main category and secondary category: for example "cs.AI" means the main category is Computer Science and the sub-category is AI, math.AG means the main category is Mathematics and the sub-category is Algebraic Geometry etc. Hence, we create three types of splits:

(a) Main category clustering Articles are only clustered based on the main category (Math, Physics, Computer Science etc.). This split evaluates coarse clustering capacity of a model.

(b) Secondary category clustering within the same main category Articles are clustered based on their secondary category, but within a given main category, for example only Math papers that need to be clustered into Algebraic Geometry, Functional Analysis, Numerical Analysis etc. This split evaluates fine-grained clustering capacity of a model, as differentiating some sub-categories can be very difficult.

(c) Secondary category clustering Articles are clustered based on their secondary category for all main categories, so the labels can be Number Theory, Computational Complexity, Astrophysics of Galaxies etc. These splits evaluate fine-grained clustering capacity, as well as multi-scale capacities i.e. a model able to both separate Maths from Physics as well as Probability from Algebraic Topology at the same time.

For every dataset, split and strategy, we select subsets of all labels and then sample articles from those labels. This yields splits with a varying amount and size of clusters.

RedditClustering (Geigle et al., 2021): Clustering of titles from 199 subreddits. Clustering of 25 splits, each with 10-50 classes, and each class with 100 - 1000 sentences

RedditClusteringP2P Dataset created for MTEB using available data from Reddit posts\(^11\). The task consists of clustering the concatenation of title+post according to their subreddit. It contains 10 splits, with 10 and 100 clusters per split and 1,000 to 100,000 posts.

StackExchangeClustering (Geigle et al., 2021) Clustering of titles from 121 stackexchanges. Clustering of 25 splits, each with 10-50 classes, and each class with 100-1000 sentences.

StackExchangeClusteringP2P Dataset created for MTEB using available data from StackExchange posts\(^12\). The task consists of clustering the concatenation of title and post according to their subreddit. It contains 10 splits, with 10 to 100 clusters and 5,000 to 10,000 posts per split.

TwentyNewsgroupsClustering\(^13\) Clustering of the 20 Newsgroups dataset, given titles of article the goal is to find the newsgroup (20 in total). Contains 10 splits, each with 20 classes, with each split containing between 1,000 and 10,000 titles.

A.2 Classification

AmazonCounterfactual (O’Neill et al., 2021) A collection of Amazon customer reviews annotated for counterfactual detection pair classification. For each review the label is either "counterfactual" or "not-counterfactual". This is a multilingual dataset with 4 available languages.

\(^9\)https://arxiv.org/help/api/
\(^10\)https://api.biorxiv.org/
\(^11\)https://huggingface.co/datasets/sentence-transformers/reddit-title-body
\(^12\)https://huggingface.co/datasets/flax-sentence-embeddings/stackexchange_title_body_jsonl
\(^13\)https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html
### Table 2: Tasks in MTEB

**AmazonPolarity** (McAuley and Leskovec, 2013) A collection of Amazon customer reviews annotated for polarity classification. For each review the label is either "positive" or "negative".

**AmazonReviews** (McAuley and Leskovec, 2013) A collection of Amazon reviews designed to aid research in multilingual text classification. For each review the label is the score given by the review between 0 and 4 (1-5 stars). This is a...
multilingual dataset with 6 available languages.

**Banking77** (Casanueva et al., 2020) Dataset composed of online banking queries annotated with their corresponding intents. For each user query the label is an intent among 77 intents like ‘activate_my_card’, ‘apple_pay’, ‘bank_transfer’, etc.  

**Emotion** (Saravia et al., 2018) Dataset of English Twitter messages with six basic emotions: anger, fear, joy, love, sadness, and surprise.  

**Imdb** (Maas et al., 2011) Large movie review dataset with labels being positive or negative.  

**MassiveIntent** (FitzGerald et al., 2022) A collection of Amazon Alexa virtual assistant utterances annotated with the associated intent. For each user utterance the label is one of 60 intents like ‘play_music’, ‘alarm_set’, etc. This is a multilingual dataset with 51 available languages.  

**MassiveScenario** (FitzGerald et al., 2022) A collection of Amazon Alexa virtual assistant utterances annotated with the associated intent. For each user utterance the label is a theme among 60 scenarios like ‘music’, ‘weather’, etc. This is a multilingual dataset with 51 available languages.  

**MTOPDomain / MTOPIntent** Multilingual sentence datasets from the MTOP (Li et al., 2020) benchmark. We refer to their paper for details.  

**ToxicConversations** Dataset from Kaggle competition\(^{14}\). Collection of comments from the Civil Comments platform together with annotations if the comment is toxic or not.  

**TweetSentimentExtraction** Dataset from Kaggle competition\(^{15}\). Sentiment classification of tweets as neutral, positive or negative.

**A.3 Pair Classification**

**SprintDuplicateQuestions** (Shah et al., 2018): Collection of questions from the Sprint community. The goal is to classify a pair of sentences as duplicates or not.

**TwitterSemEval2015** (Xu et al., 2015) Paraphrase-Pairs of Tweets from the SemEval 2015 workshop. The goal is to classify a pair of tweets as paraphrases or not.

**TwitterURLCorpus** (Lan et al., 2017) Paraphrase-Pairs of Tweets. The goal is to classify a pair of tweets as paraphrases or not.

**A.4 Bitext Mining**

**BUCC** (Zweigenbaum et al., 2016, 2017, 2018) BUCC provides big set of sentences (\(\sim 10-70k\) each) for English, French, Russian, German and Chinese, along with associated pairs annotation. The annotated pairs here corresponds to a pairs of translated sentences, i.e. a sentence and its translation in the other language.

**Tatoeba** (Research) Tatoeba provides sets of sentences (1000 sentences each) for 112 languages with annoated associated pairs. Each pair is one sentence and its translation in another language.

**A.5 Reranking**

**AskUbuntuDupQuestions**\(^{16}\) Questions from AskUbuntu with manual annotations marking pairs of questions as similar or dissimilar.

**MindSmall** (Wu et al., 2020) Large-scale English Dataset for News Recommendation Research. Ranking news article titles given the title of a news article. The idea is to recommend other news from the one you are reading.

**SciDocsRR** (Cohan et al., 2020b) Ranking of related scientific papers based on their title.

**StackOverflowDupQuestions** (Liu et al., 2018) Stack Overflow Duplicate Questions Task for questions with the tags Java, JavaScript and Python, ranking questions as duplicates or not.

**A.6 Semantic Textual Similarity (STS)**

**STS12, STS13, STS14, STS15, STS16, STS17, STS22, STSBenchmark** (Agirre et al., 2012, 2013)\(^{17181920}\) Original STS benchmark, with scores from 0 to 5. The selection of sentences includes text from image captions, news headlines and user forums. In total they contain between 1,000 and 20,000 sentences. STS12 - STS16 and

---


\(^{15}\)https://www.kaggle.com/competitions/tweet-sentiment-extraction/overview  

\(^{16}\)https://github.com/taolei87/askubuntu  

\(^{17}\)https://alt.qcri.org/semeval2014/task10/  

\(^{18}\)https://alt.qcri.org/semeval2015/task2/  

\(^{19}\)https://alt.qcri.org/semeval2016/task1/  

\(^{20}\)https://competitions.codalab.org/competitions/33835
STS Benchmark are monolingual English benchmarks. STS17 and STS22 contain cross-lingual pairs of sentences, where the goal is to assess the similarity of two sentences in different languages. STS17 has 11 language pairs (among Korean, Arabic, English, French, German, Turkish, Spanish, Italian and Dutch) and STS22 has 18 language pairs (among Arabic, English, French, German, Turkish, Spanish, Polish, Italian, Russian and Chinese).

BIOSSES\textsuperscript{21} Contains 100 sentence pairs from the biomedical field.

SICK-R (Agirre et al., 2014) Sentences Involving Compositional Knowledge (SICK) contains a large number of sentence pairs (10,000) that are lexically, syntactically and semantically rich.

A.7 Summarization

SummEval (Fabbri et al., 2020) Summaries generated by recent summarization models trained on CNN or DailyMail alongside human annotations.

A.8 Retrieval

We refer to the BEIR paper (Thakur et al., 2021), which contains description of each dataset. For MTEB, we include all publicly available datasets: ArguAna, ClimateFEVER, CQADupstack, DBPedia, FEVER, FiQA2018, HotpotQA, MSMARCO, NFCorpus, NQ, Quora, SCIDOCS, SciFact, Touche2020, TRECCovid.

B Examples

Tables 3-9 provide examples for each dataset for each task. For retrieval datasets, we refer to the BEIR paper (Thakur et al., 2021).

C Correlations

Figure 6 provides correlation heatmaps for model performance and MTEB tasks.

D Models

Table 10 provides publicly available model checkpoints used for MTEB evaluation.

E Additional results

Tables 11 until the end provide results on individual datasets of MTEB. The results are additionally available in json format on the Hugging Face Hub\textsuperscript{22} and can be inspected on the leaderboard\textsuperscript{23}.

\textsuperscript{21}https://tabilab.cmpe.boun.edu.tr/BIOSSES/DataSet.html
\textsuperscript{22}https://huggingface.co/datasets/mteb/results
\textsuperscript{23}https://huggingface.co/spaces/mteb/leaderboard
How do I know if my Franklin U772 USB Modem has a weak signal? All the home alone's watching 8 mile, "All the home alone's watching 8 mile", and women in the sciences...
Table 6: Reranking examples

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Sentence 1</th>
<th>Sentence 2</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOSSES</td>
<td>It has recently been shown that Craf is essential for Kras G12D-induced NSCLC.</td>
<td>It has recently become evident that Craf is essential for the onset of Kras-driven non-small cell lung cancer.</td>
<td>4.0</td>
</tr>
<tr>
<td>SICK-R</td>
<td>A group of children is playing in the house and there is no man standing in the background</td>
<td>A group of kids is playing in a yard and an old man is standing in the background</td>
<td>3.2</td>
</tr>
<tr>
<td>STS12</td>
<td>Nationally, the federal Centers for Disease Control and Prevention recorded 4,156 cases of West Nile, including 284 deaths.</td>
<td>There were 293 human cases of West Nile in Indiana in 2002, including 11 deaths statewide.</td>
<td>1.7</td>
</tr>
<tr>
<td>STS13</td>
<td>this frame has to do with people (the residents) residing in locations, sometimes with a co-resident.</td>
<td>inhabit or live in; be an inhabitant of;</td>
<td>2.8</td>
</tr>
<tr>
<td>STS14</td>
<td>then the captain was gone.</td>
<td>then the captain came back.</td>
<td>0.8</td>
</tr>
<tr>
<td>STS15</td>
<td>you ’ll need to check the particular policies of each publisher to see what is allowed and what is not allowed.</td>
<td>if you need to publish the book and you have found one publisher that allows it.</td>
<td>3.0</td>
</tr>
<tr>
<td>STS16</td>
<td>you do not need to worry.</td>
<td>you don ’t have to worry.</td>
<td>5.0</td>
</tr>
<tr>
<td>STS17</td>
<td>La gente muestra su afecto el uno por el otro.</td>
<td>A women giving something to other lady.</td>
<td>1.4</td>
</tr>
<tr>
<td>STS22</td>
<td>El secretario general de la Asociación Gremial de los Trabajadores del Subte y Premetro de Metrodelegados, Beto Pianelli, dijo que el Gobierno porteño debe convocar “inmediatamente” a licitación para la compra de nuevos trenes y retirar los que queden en circulación...</td>
<td>En diálogo con el servicio informativo de la Radio Pública, el ministro de Salud de la Nación, Ginés González García, habló sobre el avance del coronavirus en la Argentina y se manifestó a favor de prorrogar la cuarentena obligatoria dispuesta por...</td>
<td>1</td>
</tr>
<tr>
<td>STSBenchmark</td>
<td>A man is playing the cello.</td>
<td>A man seated is playing the cello.</td>
<td>4.25</td>
</tr>
</tbody>
</table>

Table 7: STS examples. Scores are continuous between 0 and 5 (included).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>First set sentence</th>
<th>Second set sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUCC</td>
<td>Morales remporte l’élection présidentielle de 2005 à la majorité absolue.</td>
<td>Morales went on to win the 2005 presidential election with an absolute majority.</td>
</tr>
<tr>
<td>Tatoeba</td>
<td>Chi le ha detto che Tom l’ha fatto?</td>
<td>Who told you that Tom did that?</td>
</tr>
</tbody>
</table>

Table 8: Bitext mining examples

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Human Summary</th>
<th>Machine Summary</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SummEval</td>
<td>V. Stiviano must pay back $2.6 million in gifts from Donald Sterling. Sterling’s wife claimed the ex-Clippers used the couple’s money for the gifts. The items included a Ferrari, two Bentleys and a Range Rover.</td>
<td>donald sterling, rba team last year. sterling’s wife sued for $2.6 million in gifts. sterling says he is the former female companion who has lost the. sterling has ordered v. stiviano to pay back $2.6 in in gifts after his wife sued. sterling also includes a $ 391 easter bunny costume, $ 299 and a $ 299.</td>
<td>1.7</td>
</tr>
</tbody>
</table>
Figure 6: Pearson correlations across model and task results. **Left:** Size variants of the same architecture show high correlations. **Right:** Performance on clustering and reranking correlates strongest, while summarization and classification show weaker correlation with other tasks.

<table>
<thead>
<tr>
<th>Model</th>
<th>Public Checkpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glove</td>
<td><a href="https://huggingface.co/sentence-transformers/average_word_embeddings_glove.6B.300d">https://huggingface.co/sentence-transformers/average_word_embeddings_glove.6B.300d</a></td>
</tr>
<tr>
<td>Komninos</td>
<td><a href="https://huggingface.co/sentence-transformers/average_word_embeddings_komninos">https://huggingface.co/sentence-transformers/average_word_embeddings_komninos</a></td>
</tr>
<tr>
<td>BERT</td>
<td><a href="https://huggingface.co/bert-base-uncased">https://huggingface.co/bert-base-uncased</a></td>
</tr>
<tr>
<td>SimCSE-BERT-unsup</td>
<td><a href="https://huggingface.co/princeton-nlp/unsup-simcse-bert-base-uncased">https://huggingface.co/princeton-nlp/unsup-simcse-bert-base-uncased</a></td>
</tr>
<tr>
<td>SimCSE-BERT-sup</td>
<td><a href="https://huggingface.co/princeton-nlp/sup-simcse-bert-base-uncased">https://huggingface.co/princeton-nlp/sup-simcse-bert-base-uncased</a></td>
</tr>
<tr>
<td>coCondenser-msmarco</td>
<td><a href="https://huggingface.co/sentence-transformers/msmarco-bert-co-condensor">https://huggingface.co/sentence-transformers/msmarco-bert-co-condensor</a></td>
</tr>
<tr>
<td>Contriever</td>
<td><a href="https://huggingface.co/sentence-transformers/alllenas-specter">https://huggingface.co/sentence-transformers/alllenas-specter</a></td>
</tr>
<tr>
<td>LASER2</td>
<td><a href="https://huggingface.co/sentence-transformers/LaBSE">https://huggingface.co/sentence-transformers/LaBSE</a></td>
</tr>
<tr>
<td>MPNet</td>
<td><a href="https://huggingface.co/sentence-transformers/all-mpnet-base-v2">https://huggingface.co/sentence-transformers/all-mpnet-base-v2</a></td>
</tr>
<tr>
<td>MPNet-multilingual</td>
<td><a href="https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2">https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2</a></td>
</tr>
<tr>
<td>SinGPT-5.8B-ali</td>
<td><a href="https://huggingface.co/sinennighoff/SinGPT-5.8B-weightedmean-ali-bitfit">https://huggingface.co/sinennighoff/SinGPT-5.8B-weightedmean-ali-bitfit</a></td>
</tr>
<tr>
<td>SinGPT-125M-ali</td>
<td><a href="https://huggingface.co/sinennighoff/SinGPT-125M-weightedmean-ali-bitfit">https://huggingface.co/sinennighoff/SinGPT-125M-weightedmean-ali-bitfit</a></td>
</tr>
<tr>
<td>SinGPT-1.3B-msmarco</td>
<td><a href="https://huggingface.co/sinennighoff/SinGPT-1.3B-weightedmean-msmarco-specb-bitfit">https://huggingface.co/sinennighoff/SinGPT-1.3B-weightedmean-msmarco-specb-bitfit</a></td>
</tr>
<tr>
<td>SinGPT-2.7B-msmarco</td>
<td><a href="https://huggingface.co/sinennighoff/SinGPT-2.7B-weightedmean-msmarco-specb-bitfit">https://huggingface.co/sinennighoff/SinGPT-2.7B-weightedmean-msmarco-specb-bitfit</a></td>
</tr>
<tr>
<td>SinGPT-5.8B-msmarco</td>
<td><a href="https://huggingface.co/sinennighoff/SinGPT-5.8B-weightedmean-msmarco-specb-bitfit">https://huggingface.co/sinennighoff/SinGPT-5.8B-weightedmean-msmarco-specb-bitfit</a></td>
</tr>
<tr>
<td>SinGPT-BLOOM-7.1B-msmarco</td>
<td><a href="https://huggingface.co/bigscience/singpt-bloom-71b1-msmarco">https://huggingface.co/bigscience/singpt-bloom-71b1-msmarco</a></td>
</tr>
<tr>
<td>SinGPT-BLOOM-1.7B-ali</td>
<td><a href="https://huggingface.co/bigscience-data/singpt-bloom-17b-nil">https://huggingface.co/bigscience-data/singpt-bloom-17b-nil</a></td>
</tr>
<tr>
<td>GTR-Base</td>
<td><a href="https://huggingface.co/sentence-transformers/gtr-t5-base">https://huggingface.co/sentence-transformers/gtr-t5-base</a></td>
</tr>
<tr>
<td>GTR-Large</td>
<td><a href="https://huggingface.co/sentence-transformers/gtr-t5-large">https://huggingface.co/sentence-transformers/gtr-t5-large</a></td>
</tr>
<tr>
<td>GTR-XL</td>
<td><a href="https://huggingface.co/sentence-transformers/gtr-t5-xl">https://huggingface.co/sentence-transformers/gtr-t5-xl</a></td>
</tr>
<tr>
<td>GTR XXL</td>
<td><a href="https://huggingface.co/sentence-transformers/gtr-t5-xxl">https://huggingface.co/sentence-transformers/gtr-t5-xxl</a></td>
</tr>
<tr>
<td>ST5-Base</td>
<td><a href="https://huggingface.co/sentence-transformers/sentence-t5-base">https://huggingface.co/sentence-transformers/sentence-t5-base</a></td>
</tr>
<tr>
<td>ST5-Large</td>
<td><a href="https://huggingface.co/sentence-transformers/sentence-t5-large">https://huggingface.co/sentence-transformers/sentence-t5-large</a></td>
</tr>
<tr>
<td>ST5-XL</td>
<td><a href="https://huggingface.co/sentence-transformers/sentence-t5-xl">https://huggingface.co/sentence-transformers/sentence-t5-xl</a></td>
</tr>
<tr>
<td>ST5-XXL</td>
<td><a href="https://huggingface.co/sentence-transformers/sentence-t5-xxl">https://huggingface.co/sentence-transformers/sentence-t5-xxl</a></td>
</tr>
</tbody>
</table>

Table 10: Publicly available model links used for evaluation
Table 11: All English results. The main score for each task is reported as described in Section 3.2.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Language</th>
<th>LASER2</th>
<th>LaBSE</th>
<th>MiniLM-L12-multilingual</th>
<th>MPNet-multilingual</th>
<th>SGPT-BLOOM-7.1B-msmarco</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUCC</td>
<td>de-en</td>
<td>99.21</td>
<td>99.35</td>
<td>76.00</td>
<td>78.25</td>
<td>79.58</td>
</tr>
<tr>
<td>Average mix</td>
<td></td>
<td>67.42</td>
<td>81.75</td>
<td>57.98</td>
<td>63.38</td>
<td>31.08</td>
</tr>
</tbody>
</table>

Table 12: Multilingual bitext mining results. Scores are f1.
## Table 13: Multilingual classification results. Scores are accuracy.

<table>
<thead>
<tr>
<th>Dataset Language</th>
<th>LASER2</th>
<th>LaBSE</th>
<th>MiniLM-L12-multilingual</th>
<th>MPNet-multilingual</th>
<th>SGPT-BLOOM-7.1B-msmarco</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average mix</td>
<td>42.85</td>
<td>60.77</td>
<td>54.87</td>
<td>60.39</td>
<td>54.4</td>
</tr>
<tr>
<td>Table 13: Multilingual classification results. Scores are accuracy.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>49.36</strong></td>
<td>50.07</td>
<td>64.61</td>
<td>61.96</td>
<td>64.80</td>
<td>49.36</td>
</tr>
<tr>
<td><strong>Average mix</strong></td>
<td>50.07</td>
<td>64.61</td>
<td>61.96</td>
<td>64.80</td>
<td>49.36</td>
</tr>
<tr>
<td><strong>50.07</strong></td>
<td>50.07</td>
<td>64.61</td>
<td>61.96</td>
<td>64.80</td>
<td>49.36</td>
</tr>
<tr>
<td><strong>50.07</strong></td>
<td>50.07</td>
<td>64.61</td>
<td>61.96</td>
<td>64.80</td>
<td>49.36</td>
</tr>
<tr>
<td><strong>50.07</strong></td>
<td>50.07</td>
<td>64.61</td>
<td>61.96</td>
<td>64.80</td>
<td>49.36</td>
</tr>
<tr>
<td><strong>50.07</strong></td>
<td>50.07</td>
<td>64.61</td>
<td>61.96</td>
<td>64.80</td>
<td>49.36</td>
</tr>
<tr>
<td>Dataset</td>
<td>Language</td>
<td>Komninos</td>
<td>LASER2</td>
<td>LaBSE</td>
<td>MmLM-L12-multilingual</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>----------</td>
<td>--------</td>
<td>-------</td>
<td>------------------------</td>
</tr>
<tr>
<td>STS17</td>
<td>ko-ko</td>
<td>2.54</td>
<td>70.52</td>
<td>71.32</td>
<td>77.03</td>
</tr>
<tr>
<td>STS17</td>
<td>ar-ar</td>
<td>13.78</td>
<td>67.47</td>
<td>69.07</td>
<td>79.16</td>
</tr>
<tr>
<td>STS17</td>
<td>en-ar</td>
<td>9.08</td>
<td>65.05</td>
<td>74.51</td>
<td>81.22</td>
</tr>
<tr>
<td>STS17</td>
<td>en-de</td>
<td>-3.11</td>
<td>66.66</td>
<td>73.85</td>
<td>84.22</td>
</tr>
<tr>
<td>STS17</td>
<td>en-tr</td>
<td>-0.45</td>
<td>70.05</td>
<td>72.07</td>
<td>76.74</td>
</tr>
<tr>
<td>STS17</td>
<td>es-en</td>
<td>-8.18</td>
<td>55.30</td>
<td>65.71</td>
<td>84.44</td>
</tr>
<tr>
<td>STS17</td>
<td>es-es</td>
<td>48.23</td>
<td>79.67</td>
<td>80.83</td>
<td>85.56</td>
</tr>
<tr>
<td>STS17</td>
<td>fr-en</td>
<td>5.81</td>
<td>70.82</td>
<td>76.98</td>
<td>76.59</td>
</tr>
<tr>
<td>STS17</td>
<td>it-en</td>
<td>3.64</td>
<td>70.98</td>
<td>76.99</td>
<td>82.35</td>
</tr>
<tr>
<td>STS17</td>
<td>nl-en</td>
<td>-0.44</td>
<td>68.12</td>
<td>75.22</td>
<td>81.71</td>
</tr>
<tr>
<td>STS22</td>
<td>de</td>
<td>33.04</td>
<td>25.69</td>
<td>48.58</td>
<td>44.64</td>
</tr>
<tr>
<td>STS22</td>
<td>es</td>
<td>48.53</td>
<td>54.92</td>
<td>63.18</td>
<td>56.56</td>
</tr>
<tr>
<td>STS22</td>
<td>pl</td>
<td>12.47</td>
<td>18.34</td>
<td>39.30</td>
<td>33.74</td>
</tr>
<tr>
<td>STS22</td>
<td>tr</td>
<td>47.38</td>
<td>36.97</td>
<td>58.15</td>
<td>53.39</td>
</tr>
<tr>
<td>STS22</td>
<td>ar</td>
<td>32.42</td>
<td>42.57</td>
<td>57.67</td>
<td>46.2</td>
</tr>
<tr>
<td>STS22</td>
<td>ru</td>
<td>19.44</td>
<td>39.24</td>
<td>57.49</td>
<td>57.08</td>
</tr>
<tr>
<td>STS22</td>
<td>zh</td>
<td>4.78</td>
<td>49.41</td>
<td>63.02</td>
<td>58.75</td>
</tr>
<tr>
<td>STS22</td>
<td>fr</td>
<td>49.43</td>
<td>58.61</td>
<td>77.95</td>
<td>70.55</td>
</tr>
<tr>
<td>STS22</td>
<td>de-en</td>
<td>28.65</td>
<td>32.35</td>
<td>50.14</td>
<td>52.65</td>
</tr>
<tr>
<td>STS22</td>
<td>es-en</td>
<td>26.97</td>
<td>54.34</td>
<td>71.86</td>
<td>67.33</td>
</tr>
<tr>
<td>STS22</td>
<td>it</td>
<td>57.77</td>
<td>60.31</td>
<td>72.22</td>
<td>55.22</td>
</tr>
<tr>
<td>STS22</td>
<td>pl-en</td>
<td>45.55</td>
<td>53.63</td>
<td>69.41</td>
<td>69.02</td>
</tr>
<tr>
<td>STS22</td>
<td>zh-en</td>
<td>14.05</td>
<td>46.19</td>
<td>64.02</td>
<td>65.71</td>
</tr>
<tr>
<td>STS22</td>
<td>es-it</td>
<td>41.10</td>
<td>42.21</td>
<td>69.69</td>
<td>47.67</td>
</tr>
<tr>
<td>STS22</td>
<td>de-fr</td>
<td>14.77</td>
<td>37.41</td>
<td>53.28</td>
<td>51.73</td>
</tr>
<tr>
<td>STS22</td>
<td>de-pl</td>
<td>11.21</td>
<td>15.67</td>
<td>58.69</td>
<td>44.22</td>
</tr>
<tr>
<td>STS22</td>
<td>fr-pl</td>
<td>39.44</td>
<td>39.44</td>
<td>61.98</td>
<td>50.71</td>
</tr>
<tr>
<td><strong>Average mix</strong></td>
<td>22.14</td>
<td>51.55</td>
<td>65.67</td>
<td>64.23</td>
<td>67.71</td>
</tr>
</tbody>
</table>

Table 14: Multilingual STS Results. Scores are Spearman correlations of cosine similarities.