ChemTEB: Chemical Text Embedding Benchmark, an Overview of Embedding Models Performance & Efficiency on a Specific Domain

Ali Shiraee Kasmaee[1,2,*,†], Mohammad Khodadad[1,2,†], Mohammad Arshi Saloot[2], Nick Sherck[3], Stephen Dokas[3], Hamidreza Mahyar[1], Soheila Samiee[2]

[1]Department of Computational Science and Engineering, McMaster University, Canada
[2]BASF Canada Inc., Canada
[3]BASF Corporation, USA
{shiraeea, khodam3, mahyarh}@mcmaster.ca
{mohammad.arshi-saloot, nicholas.sherck, stephen.dokas, soheila.samiee}@basf.com
*Corresponding Author: shiraeea@mcmaster.ca
†Equal Contribution

Abstract

Recent advancements in language models have started a new era of superior information retrieval and content generation, with embedding models playing an important role in optimizing data representation efficiency and performance. While benchmarks like the Massive Text Embedding Benchmark (MTEB) have standardized the evaluation of general domain embedding models, a gap remains in specialized fields such as chemistry, which require tailored approaches due to domain-specific challenges. This paper introduces a novel benchmark, the Chemical Text Embedding Benchmark (ChemTEB), designed specifically for the chemical sciences. ChemTEB addresses the unique linguistic and semantic complexities of chemical literature and data, offering a comprehensive suite of tasks on chemical domain data. Through the evaluation of 34 open-source and proprietary models using this benchmark, we illuminate the strengths and weaknesses of current methodologies in processing and understanding chemical information. Our work aims to equip the research community with a standardized, domain-specific evaluation framework, promoting the development of more precise and efficient NLP models for chemistry-related applications. Furthermore, it provides insights into the performance of generic models in a domain-specific context. ChemTEB comes with open-source code [1] and data [2], contributing further to its accessibility and utility.

1 Introduction

Deep learning and natural language processing have improved significantly, highlighting the importance of learning text representation to understand semantic similarity. This is a key part of text mining, search, retrieval, and other similar tasks. In the last decade, many promising models have been developed to meet this need. Earlier models, such as GloVe [1] and Word2vec [2] focused on word embedding but lacked context awareness. More recent models have adopted transformer archi-

---

[1]https://github.com/basf/chemteb
[2]https://huggingface.co/BASF-AI

tecture to incorporate context into token embeddings [3]. Each model proposes unique architectural features, parameter counts, context lengths, and pretraining methods.

BERT was among the first models to employ transformer architecture and self-supervised training for embedding models [3, 4]. Subsequently, variants of BERT were introduced to enhance performance, such as ROBERTA [5], or to facilitate domain adaptation for specific areas of interest, such as SciBERT for the sciences [6]. Initially, a common approach to achieve a single embedding vector for a text corpus – i.e. pooling – was to average the output layer or use the [CLS] token; however, it has been found to be less effective in grasping semantically meaningful embeddings. Sentence-BERT [7] is a promising fine-tuned variant that leverages a Siamese bi-encoder and triplet loss [8] to achieve competitive performance in semantic representation learning, making it highly suitable for embedding tasks. Models such as E5 [9] and Nomic embed [10] have integrated contrastive learning into the pretraining process in an effort to help the models better distinguish between similar and dissimilar samples, thereby enhancing implementation efficiency. The BGE model family [11] stands out by integrating an MAE-inspired pre-training approach [12] with contrastive learning, utilizing large batch sizes to enhance embedding quality. M3-embedding [13] has emphasized multi-functionality, multi-granularity, and multi-linguality in its embedding model to further improve the performance. Furthermore, OpenAI, Cohere, and Amazon have introduced their proprietary models for embedding, continually expanding the list of available models.

The progress in natural language processing (NLP) has influenced a wide range of scientific domains, including biology, medicine, and physics. These advancements have enabled researchers to extract, analyze, and interpret vast amounts of textual data with unprecedented accuracy and efficiency. Embedding models are crucial for solving complex tasks across these domains. These models transform high-dimensional data into dense vector spaces, capturing semantic relationships essential for applications such as chemical literature mining and even molecular property prediction. The rise of Retrieval-Augmented Generation (RAG) architectures [14], which combine language models with external knowledge retrieval systems, offers new opportunities for these tasks. RAG enhances embedding-based applications by allowing dynamic access to domain-specific knowledge, making them highly effective for tasks that require both deep learning and external information sources. With the increased application of such methods, the requirement of having efficient embedding models is raised in the industry. While general NLP benchmarks like the Massive Text Embedding Benchmark (MTEB) have been instrumental in standardizing model evaluations across a variety of tasks [15], they fall short when applied to chemistry-specific tasks. For example, the linguistic and semantic nuances inherent in chemical literature are often overlooked by models trained and evaluated on general datasets. This gap underscores the need for a specialized benchmark tailored to the domain of chemistry, where precision and context are of paramount importance. Improved NLP models have the potential to revolutionize various aspects of chemistry, such as automated literature reviews, chemical synthesis planning, patent analysis and even contribute to further improvement of Autonomous Agents in Chemistry [16, 17].

To bridge this gap, we introduce a novel benchmark explicitly tailored for the chemical sciences, named Chemical Text Embedding Benchmark (ChemTEB). This benchmark offers a comprehensive suite of tasks, ranging from chemical text classification to bitext mining of natural language and SMILES representation of chemical compounds. Our benchmark aims to provide a robust, domain-specific evaluation framework, promoting the development of more precise and efficient NLP models for applications related to chemistry. Accompanied by an open-source package and data, ChemTEB enables straightforward evaluation of any model and facilitates the effortless incorporation of new tasks and datasets.

2   Related Work

To the best of our knowledge, there is no previous benchmark on chemical embedding models. The closest previous benchmarks were either generic NLP benchmarks on information retrieval or semantic similarity; or chemical benchmarks on generative large language models, machine learning methods on molecular data or Graph Neural Networks (GNNs). In the following, more details on these benchmarks are provided.

2.1 NLP Benchmarks

The evaluation of sentence embeddings and general language understanding has seen significant advancements, driven by various benchmarks and toolkits designed to assess models across a diverse set of tasks. One prominent toolkit is SentEval [18], widely used for evaluating sentence embeddings on tasks such as semantic textual similarity, paraphrase detection, and sentiment analysis. SentEval is complemented by the GLUE Benchmark [19], which assesses models on a broader range of natural language processing (NLP) tasks, including sentence embeddings, to measure language understanding. Building on this success, SuperGLUE [20] takes evaluation a step further, introducing more complex tasks that challenge models to push beyond existing performance levels. For more targeted evaluation, the STS Benchmark [21] focuses on measuring semantic similarity between sentence pairs, which is critical for determining the quality of sentence embeddings. In the context of information retrieval, BEIR [22] provides a heterogeneous benchmark, incorporating datasets that evaluate models on tasks like retrieval, ranking, and embedding quality.

Cross-lingual performance is another essential dimension in embedding evaluation, tackled by benchmarks like XTREME [23], which tests models on classification, retrieval, and sentence similarity tasks across multiple languages. XGLUE [24], a multilingual extension of the GLUE framework, offers further evaluation on a wide range of NLP tasks across different languages. Meanwhile, the TREC benchmarks [25] have been a long-standing standard for testing information retrieval systems, particularly for embedding models in retrieval scenarios. MTEB [15] broadens the scope by evaluating embeddings across a variety of language-understanding tasks and domains, focusing on scalability and adaptability. Additionally, specialized NLP tasks like natural language inference and paraphrase recognition are tackled by datasets such as MultiNLI [26] and PAWS [27], which offer challenging benchmarks for testing the depth of understanding in sentence embeddings. For Chinese NLP tasks, CLUE [28] serves as a counterpart to GLUE, addressing the unique challenges of the Chinese language. Despite the breadth of these benchmarks, one notable gap remains; the absence of a comparable evaluation framework for chemistry-related language models.

2.2 Chemistry Benchmarks

While there have been attempts to apply NLP in chemistry, such as ChemDataExtractor [29] and natural language processing techniques to extract information on the properties and functionalities of energetic materials from large text corpora [30], these efforts are often limited by the lack of a standardized evaluation framework. Existing resources like the ChEMBL database [31] offer valuable data but do not provide the comprehensive, task-oriented benchmarking necessary to drive significant advances in chemical NLP. The ChemNLP [32] library advances the application of NLP in chemistry by providing curated datasets from arXiv and PubChem, along with tools for visualization, analysis, and task execution tailored to materials chemistry. However, there is no proper task or code for the evaluation of developed models.

After the rise of large generative models, some benchmark datasets and evaluation frameworks have been developed to assess the capabilities of large language models (LLMs) and machine learning techniques in the fields of materials science and chemistry. MaterialBENCH [33], ChemBench [34], and ChemLLMBench [35] are some of these studies. However, none of them provided tasks and data on *embedding* models evaluation.

3 ChemTEB

In this work, we leverage a diverse set of datasets collected to evaluate embedding models across diverse tasks, including Classification, Pair Classification, Clustering, Retrieval, and Bitext Mining. The data sources used are PubChem [36], English Wikipedia, BeIR [37], CoconutDB [38], and Safety Data Sheets [39] each offering unique and complementary information critical for evaluating the performance of NLP models in chemistry. All tasks and datasets are either designed or validated by domain experts, i.e. chemists.

3.1 Tasks

We present a variety of benchmarks designed to evaluate different aspects of natural language and chemical data processing. Each benchmark focuses on a specific task, and in this section, we provide

an overview of the task, the data sources used for its collection, and the evaluation process. These benchmarks provide a comprehensive set of evaluation tasks using diverse datasets tailored to different modeling approaches. Table 1 provides a summary of datasets and statistics associated with them.

Table 1: Datasets summary. This table provides an overview of the datasets used across different tasks, including the dataset names from Hugging Face, the original data sources, and the distribution of sample sizes. The distribution is represented through key statistical measures: 5th percentile, median, and 95th percentile of the number of tokens

Task		HuggingFace Name	Data Source	#Samples	Sequence Lengths (tokens [3])		
					5th Percentile	Median	95th Percentile
Classification	1	WikipediaEasy10Classification	Wikipedia	2105	42	178	612.4
	2	WikipediaEasy5Classification	Wikipedia	1164	43	171.5	547.85
	3	WikipediaMedium5Classification	Wikipedia	617	39	137	563.6
	4	WikipediaMedium2CrystallographyVsChromatographyTitrationpHClassification	Wikipedia	1451	41.5	175	658.5
	5	WikipediaMedium2BioluminescenceVsNeurochemistryClassification	Wikipedia	486	42	158	574.25
	6	WikipediaEZ2Classification	Wikipedia	58921	41	164	590
	7	WikipediaHard2BioluminescenceVsLuminescenceClassification	Wikipedia	410	41	148.5	579.3
	8	WikipediaEasy2GeneExpressionVsMetallurgyClassification	Wikipedia	5741	42	175	630
	9	WikipediaEasy2GreenhouseVsEnantiopureClassification	Wikipedia	1136	34	139.5	513
	10	WikipediaEZ10Classification	Wikipedia	43146	41	165	582
	11	WikipediaHard2SaltsVsSemiconductorMaterialsClassification	Wikipedia	491	38.5	141	447.5
	12	WikipediaEasy2SolidStateVsColloidalClassification	Wikipedia	2216	42	151	532
	13	WikipediaMedium2ComputationalVsSpectroscopistsClassification	Wikipedia	1101	38	155	639
	14	WikipediaHard2IsotopesVsFissionProductsNuclearFissionClassification	Wikipedia	417	43.8	209	706.4
	15	WikipediaEasy2SpecialClassification	Wikipedia	1312	35.55	133	465
	16	SDSGlovesClassification	Safety Data Sheets	8000	498	1071	1871
	17	SDSEyeProtectionClassification	Safety Data Sheets	8000	492	1060	1876
BitextMining	18	CoconutSMILES2FormulaBM	CoconutDB	8000	6	11	150
	19	PubChemSMILESIsoTitleBM	PubChem	14140	4	22	93
	20	PubChemSMILESIsoDescBM	PubChem	14140	12	45	134
	21	PubChemSMILESCanonTitleBM	PubChem	30914	3	12	43
	22	PubChemSMILESCanonDescBM	PubChem	30914	8	24	109
Retrieval	23	ChemHotpotQARetrieval	HotpotQA	10275	19	71	183
	24	ChemNQRetrieval	Natural Questions	22960	13	81	231
Clustering	25	WikipediaMedium5Clustering	Wikipedia	617	39	137	563.6
	26	WikipediaEasy10Clustering	Wikipedia	2105	42	178	612.4
PairClassification	27	WikipediaAIParagraphsParaphrasePC	Wikipedia	5408	28	104	354
	28	CoconutSMILES2FormulaPC	CoconutDB	8000	6	11	108
	29	PubChemAISentenceParaphrasePC	PubChem	4096	9	20	59
	30	PubChemSMILESCanonTitlePC	PubChem	4096	4	16	30
	31	PubChemSynonymPC	PubChem	4096	3	8	38
	32	PubChemSMILESCanonDescPC	PubChem	4096	12	23	105
	33	PubChemSMILESIsoDescPC	PubChem	4096	12	48	125
	34	PubChemSMILESIsoTitlePC	PubChem	4096	4	35	70
	35	PubChemWikiParagraphsPC	PubChem	4096	8	66	235

Classification: Each task consists of a dataset with a textual field and associated labels. A logistic regression classifier is used on top of an embedding model, which is first trained on the training split of the dataset. The performance is then evaluated on the test dataset using the F1 score. We have used two main sources to construct the datasets in this category: first, chemistry-related English Wikipedia articles classified into various chemistry subfields, and second, Safety Data Sheets (SDS) [39], which are detailed documents providing essential information on the properties and hazards of chemicals, ensuring user safety and compliance with regulatory standards.

Clustering tasks involve grouping related text pieces into meaningful clusters based on their embeddings. Similar to the classification tasks, clustering datasets are also constructed from chemistry-related English Wikipedia articles, where section texts are clustered into various chemistry subfields. A mini-batch k-means model with a batch size of 32 is trained on the text pieces. We have used V-measure [40] to assess and report the performance.

Pair classification tasks involve determining whether two textual samples are related and assigning a binary label to them. In the chemistry domain, this relationship could be whether two texts refer to the same chemical entity, or matching compound titles or their descriptions with their SMILES strings (Simplified Molecular Input Line Entry System). In this task, a pair of text is embedded using an embedding model, their distance is calculated (using metrics such as cosine similarity, Euclidean distance, Manhattan distance and dot product), and the best binary threshold for each metric is determined. The F1 score is then calculated across all metrics, with the maximum F1 score reported as the main score. The datasets for these tasks are constructed from databases containing chemical product information, such as PubChem [36] and COCONUT [38].

Bitext Mining focuses on matching pairs of text that are translations or semantic equivalents of each other, by performing a semantic similarity search between a list of query embeddings and a list of corpus embeddings. We have used data sources such as PubChem [36] and COCONUT [38] and

---

[3]All tokenizations are done with OpenAI's *cl100k_base* tokenizer via tiktoken python package.

matched the SMILES of chemical entities with their titles, descriptions and chemical formulas. We have employed f1 score to evaluate the performance of the models.

In **Retrieval** tasks, the model's ability to retrieve relevant records based on a query is evaluated. Each dataset in this category consists of a list of queries and documents, along with the mapping between relevant queries and documents. A model is used to embed all the texts, and the relevant documents are retrieved based on the cosine similarity between embeddings. We have used a chemistry-related subset of the Natural Questions [41] and HotpotQA [42] datasets and employed nDCG@10 as the main evaluation metric.

3.2 Embedding Models

In this study, a total of 34 embedding models have been evaluated in the ChemTEB benchmark. This includes 27 open-source models and 7 proprietary models. More detailed information about the evaluated models can be found in section S2 of supplementary materials, with Table S1 comparing these models based on their characteristics.

3.3 Ranking Process for Model Performance

The models are ranked based on their performance across datasets in each category of tasks. In the first step, the arithmetic mean of the performance metrics is computed for each task to summarize the performance over tasks. In the next step, an overall score was calculated using the Reciprocal Rank Fusion (RRF) method [43]:

$$\text{RRF}_{\text{score}}(m) = \sum_{d \in \text{Datasets}} \frac{1}{k + r_d(m)}$$

where $r_d(m)$ is the rank of model m for dataset d, and $k = 10$ ensures all models retain weight. Summing RRF scores across datasets gives an aggregate score for each model. Higher RRF scores reflect a better overall ranking of the model.

4 Results

4.1 Models Performance

Table 2 summarizes the average performance of each model in each category of tasks and the overall performance, which is presented with RRF_{score} (see section 3.3 for more details). From **model** perspective, there is no single model which outperforms others in all tasks, but in general proprietary models provide a better performance compared to open-source models. *OpenAI-text embedding 3-large* provided the best results in 3 out of 5 task categories, ranked first among evaluated models. Among open-source models, *Nomic Embedding v1.5* showed the best overall performance, ranking second after OpenAI large embedding model (see the detailed ranking of all models on each sub-task in Supplementary Table S3).

From **Task** category perspective, models generally performed better in classification tasks, and the worst performance has been observed in bitext mining. As explained earlier in section 3.1 bitext mining was designed on translation between SMILES representation of chemical compounds and either their title or description. The poor performance of the models in this task can be attributed to the fact that none of the general-purpose models are trained on a modality such as SMILES code. As a result, they fail to grasp the semantic relationships between different SMILES strings, leading to suboptimal performance in this specific task. Retrieval, clustering, and pair classification, were in 2nd to 4th rank respectively, in terms of difficulty level and overall performance of models.

The models evaluated in this benchmark share many features, such as architecture and training methods. For better evaluation of the impact of models' characteristics in their performance in each category of tasks, we grouped these models based on their similarities into eight families: (i) BERT Family, (ii) Nomic embedding family, (iii) SBERT family, (ix) E5 family, (x) BGE family, (xi) OpenAI family, (xii) Amazon family, and (xiii) Cohere family. Figure 1 illustrates the distribution of each model family performance over all datasets in each category of tasks using the Kernel Density Estimation (KDE - see supplementary section S4.1 for more details).

Table 2: Summary of models performance. This table provides a comprehensive comparison of models across several key tasks: text classification (macro F1-score), bitext mining (F1-score), text retrieval (nDCG@10), text clustering (F1-score), pair classification (maximum F1-score), and an overall score (Reciprocal Rank Fusion). The models are grouped into two categories—open-source and proprietary—for easier distinction. In each category, the best-performing model is underscored, while the overall best-performing model across all categories is highlighted in bold.

|	Classification (Macro F1)	Bitext Mining (F1)	Retrieval (nDCG@10)	Clustering (V-measure)	Pair Classification (Max F1)	Final Score (RRF)
BERT	0.72±0.04	0.0±0.0	0.28±0.02	0.2±0.03	0.41±0.05	0.122
SciBERT	0.71±0.04	0.0002±0.0	0.2±0.03	0.18±0.02	0.43±0.05	0.122
MatSciBERT	0.7±0.04	0.0003±0.0001	0.11±0.02	0.21±0.03	0.41±0.05	0.122
Chemical BERT	0.68±0.04	0.0003±0.0	0.17±0.01	0.13±0.02	0.42±0.05	0.120
Nomic BERT	0.67±0.04	0.0001±0.0	0.05±0.0	0.22±0.03	0.38±0.04	0.118
Nomic Embedding v1	0.77±0.04	0.0023±0.0002	0.72±0.02	0.46±0.03	**0.55±0.06**	0.285
Nomic Embedding v1.5	0.78±0.04	0.0026±0.0002	0.75±0.02	0.5±0.04	**0.55±0.06**	0.339
SBERT - all Mini LM L6.v2	0.78±0.03	0.0015±0.0002	0.61±0.01	0.36±0.02	0.54±0.06	0.232
SBERT - all Mini LM L12.v2	0.77±0.04	0.0013±0.0001	0.58±0.0	0.34±0.01	0.54±0.06	0.201
SBERT - all MPNET-base.v2	0.78±0.04	0.001±0.0001	0.56±0.0	0.5±0.03	0.54±0.06	0.239
SBERT - multi-qa-mpnet-base.v1	0.74±0.04	0.0009±0.0001	0.56±0.01	0.42±0.04	0.54±0.06	0.185
E5 - small	0.75±0.03	0.0015±0.0001	0.69±0.02	0.12±0.02	0.48±0.05	0.166
E5 - base	0.76±0.04	0.0019±0.0001	0.68±0.01	0.34±0.05	0.49±0.05	0.192
E5 - large	0.77±0.04	0.0029±0.0002	0.7±0.01	0.51±0.04	0.5±0.05	0.290
E5 - small v2	0.76±0.03	0.0012±0.0001	0.69±0.01	0.19±0.03	0.46±0.05	0.165
E5 - base v2	0.76±0.04	0.0016±0.0001	0.68±0.01	0.38±0.05	0.47±0.05	0.178
E5 - large v2	0.76±0.04	0.0022±0.0002	0.73±0.01	0.33±0.05	0.48±0.05	0.214
E5 - Multilingual small	0.74±0.04	0.0018±0.0001	**0.76±0.01**	0.17±0.01	0.47±0.05	0.207
E5 - Multilingual base	0.75±0.04	0.0022±0.0001	0.68±0.0	0.48±0.03	0.47±0.05	0.196
E5 - Multilingual large	0.74±0.04	0.0026±0.0002	0.67±0.0	0.3±0.05	0.48±0.05	0.187
BGE - small en	0.78±0.04	0.0012±0.0001	0.52±0.04	0.27±0.03	0.48±0.05	0.160
BGE - base en	0.77±0.04	0.0019±0.0003	0.59±0.03	0.44±0.05	0.48±0.05	0.186
BGE - large en	0.78±0.04	0.0016±0.0001	0.44±0.06	0.45±0.05	0.49±0.05	0.191
BGE - small en v1.5	0.78±0.03	0.0013±0.0001	0.63±0.03	0.25±0.04	0.48±0.05	0.180
BGE - base en v1.5	0.77±0.04	0.0018±0.0001	0.69±0.02	0.47±0.05	0.49±0.05	0.219
BGE - large en v1.5	0.78±0.04	0.0019±0.0001	0.67±0.02	0.39±0.06	0.5±0.05	0.224
BGE - Multilingual - M3	0.76±0.03	0.0012±0.0002	0.68±0.02	0.45±0.05	0.47±0.06	0.176
OpenAI - Text embedding 3 - small	0.78±0.04	0.0027±0.0003	0.65±0.01	0.49±0.05	0.5±0.05	0.273
OpenAI - Text embedding 3 - large	0.8±0.04	**0.0062±0.0006**	0.71±0.01	**0.6±0.03**	0.53±0.05	**0.384**
OpenAI - Text embedding - Ada - 02	0.78±0.04	0.0035±0.0002	0.66±0.02	0.52±0.04	0.49±0.05	0.279
Amazon - Titan Text Embedding v2	0.77±0.03	0.0024±0.0002	0.62±0.0	0.49±0.04	0.49±0.05	0.224
Amazon - Titan Embedding G1 Text	0.81±0.03	0.0032±0.0003	0.6±0.02	0.45±0.06	0.49±0.05	0.285
Cohere - Embed English V3	0.81±0.03	0.0012±0.0	0.49±0.04	0.55±0.02	0.53±0.06	0.278
Cohere - Embed Multilingual V3	0.8±0.03	0.0024±0.0001	0.49±0.04	0.53±0.03	0.53±0.06	0.281
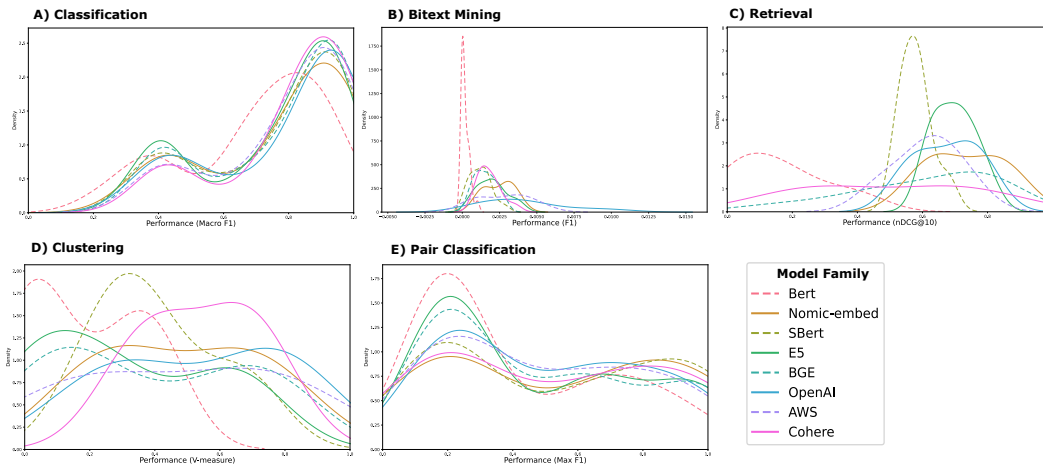
Figure 1: Distribution plots for five categories of tasks. The KDE plots show the probability density functions, where the x-axis represents the range of predicted values (performance distribution over tasks of each category and models of each family) and the y-axis represents the estimated density. Each colored line corresponds to a unique model family, enabling a clear visual comparison of their value distributions.

4.2 Models Efficiency

The evaluated models in this benchmark vary in terms of architecture, data size, model size, speed, and performance, among other factors. Depending on specific requirements, one model may be more suitable than another, but making an informed decision requires comparing models across multiple features simultaneously. To facilitate this, in Figure 2, we visualized the speed (in the pair classification task), model size, embedding size, and RRF score for each model. For proprietary models, size information was not available.

The visualization highlights the diverse performance of the models, each with its own advantages and disadvantages. A noticeable trend is that slower models tend to be larger, with bigger embedding sizes, and generally offer higher performance. On one end of the spectrum, *OpenAI - Text Embedding 3 - Large* achieved the highest RRF score but exhibited very low speed. In contrast, *SBERT - All Mini LM L6.v2* was both the smallest and fastest model, though with lower performance. Interestingly, *BERT-based* models had the lowest RRF scores and relatively slower speeds, clearly distinguishing them from other models. Notably, the open-source *Nomic Embedding v1.5* demonstrated a good balance between high speed and strong performance.
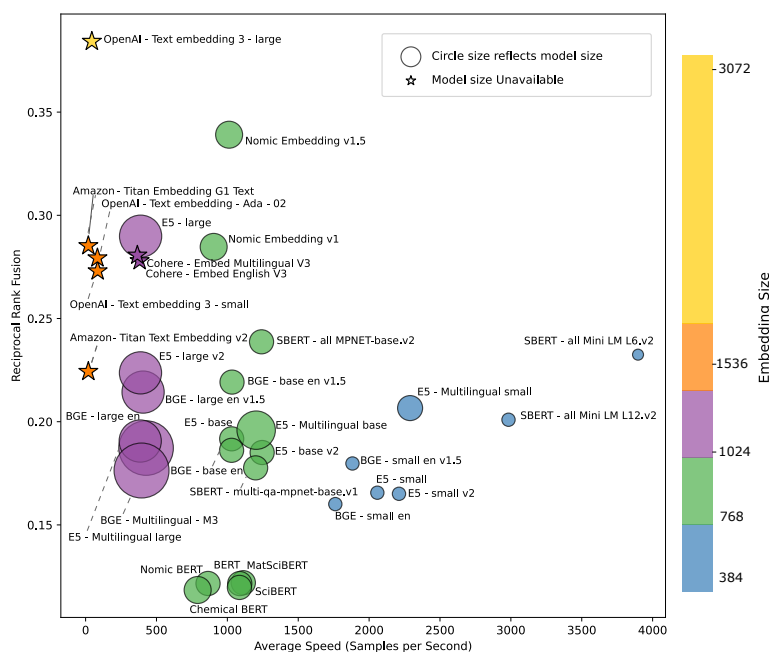

Figure 2: Summary of evaluated models in terms of efficiency. All evaluated models are depicted in the form of (i) circles (with circle size being proportional to the number of parameters) for open-source models, and (ii) stars for proprietary models. The color of the depicted models reflects their embedding dimension. The x-axis denotes the averaged inference speed (embedded samples/sec) calculated over seven pair classification tasks (tasks 29 - 35 in table 1) conducted on a V100 GPU machine.

4.3 Domain Adaptation

To the best of our knowledge, the only existing embedding models specifically adapted to the chemical domain are MatSciBERT [44], and ChemicalBERT (from Recobo[4]). SciBERT [6], pre-trained on scientific data, is also more relevant to the chemistry domain compared to generic models. Within the BERT family, these domain-adapted models outperformed BERT-base in the bitext mining task, the most challenging in-domain task requiring partial knowledge of SMILES codes. However, we did not observe a consistent significant improvement in other tasks, except for SciBERT's superior performance in pair classification.

---

[4]recobo/chemical-bert-uncased

Contrarily, outside the BERT family, these domain-adapted models performed considerably worse in most task categories, as evidenced by their joint lowest RRF-score for ranking (refer to the Table S3 in supplementary materials for their detailed ranking per task category). This could be attributed to their belonging to the BERT Family with only Masked Language Modeling (MLM) pretraining. Based on the evaluations and observations, it appears that the contrastive objective and architectural improvements introduced post-BERT for enhanced semantic representation contribute more to the performance in specific domains than domain adaptation of weaker architectures. This finding encourages the research community to move away from relying on older domain-adapted models and instead continue developing domain-specific models with newer more efficient architectures.
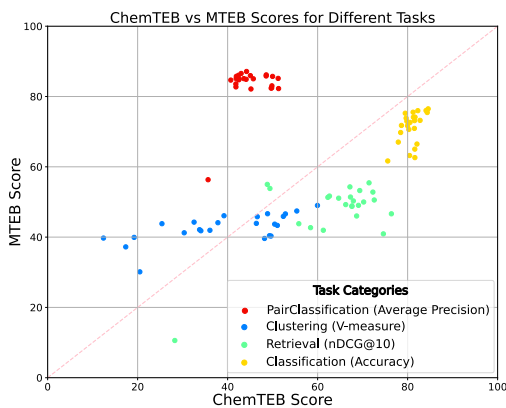

Figure 3: Comparison of model performance on ChemTEB and MTEB benchmarks across different tasks. Each point represents a model from the intersection of those tested and those on the MTEB leaderboard as of the date. The figure highlights variations in task difficulty and domain specificity.

Figure 3 illustrates the performance of various models on both ChemTEB and MTEB benchmarks, organized by task categories. Similar metrics to the MTEB leaderboard have been used here to have comparable results. In pair classification tasks, specifically curated for chemistry, a significant decline in the average precision of models in ChemTEB tasks compared to MTEB, highlights the lack of domain expertise in evaluated models. Clustering tasks reveal that ChemTEB has more discriminating datasets, as evidenced by the variability in scores. For retrieval tasks, we observed a similar trend in terms of more variability, potentially contributed by the specialized chemical context, but the performance of each model is mostly better in ChemTEB compared to MTEB. In classification tasks, ChemTEB reflects the better performance of models with less variability in results, suggesting that these tasks may be easier compared to those in MTEB. This difference is likely due to the general nature of the Wikipedia documents used in ChemTEB classification problems. (Refer to supplementary figure S1 for details of each model performance in these two benchmarks.) These observations emphasize the influence of domain adaptation on model performance and underscore the necessity for tailored approaches in specialized fields like chemistry.

5 Conclusion

In conclusion, we have addressed a significant gap in the field of text embeddings evaluation by developing ChemTEB, a unique, open-source benchmark specifically designed for the challenges of chemical language and data. This novel tool allows for a comprehensive evaluation of both open-source and proprietary models, offering a standardized measure of performance in the chemistry domain. ChemTEB provides an invaluable opportunity to assess different models, identify their synergies, and pinpoint areas where innovative solutions can lead to more efficient and high-performing tools. Its model-agnostic nature ensures that it can easily evaluate any model or incorporate new data, making it a versatile addition to the open-source benchmark repertoire.

The results of our study highlight the critical importance of developing stronger, domain-adapted models for improved representation of domain-specific data in the field of chemistry. Therefore, our work not only contributes a valuable tool to the research community but also emphasizes the need for continued innovation in the development of domain-specific models.

Acknowledgments and Disclosure of Funding

References

[1] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[2] Tomas Mikolov. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[3] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[4] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[6] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[7] N Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[8] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[9] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

[10] Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*, 2024.

[11] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649, 2024.

[12] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. *arXiv preprint arXiv:2205.12035*, 2022.

[13] Multi-Linguality Multi-Functionality Multi-Granularity. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

[14] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[15] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022.

[16] Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*, 2023.

[17] Mayk Caldas Ramos, Christopher J Collison, and Andrew D White. A review of large language models and autonomous agents in chemistry. *arXiv preprint arXiv:2407.01603*, 2024.

[18] Alexis Conneau and Douwe Kiela. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*, 2018.

[19] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[20] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[21] Daniel Cer, Mona Diab, Eneko E Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and cross-lingual focused evaluation. In *The 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, 2017.

[22] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models, 2021.

[23] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR, 2020.

[24] Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv preprint arXiv:2004.01401*, 2020.

[25] Ian Soboroff. Overview of trec 2021. In *TREC*, 2021.

[26] Seonhoon Kim, Inho Kang, and Nojun Kwak. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6586–6593, 2019.

[27] Yuan Zhang, Jason Baldridge, and Luheng He. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*, 2019.

[28] Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. Clue: A chinese language understanding evaluation benchmark. *arXiv preprint arXiv:2004.05986*, 2020.

[29] Matthew C Swain and Jacqueline M Cole. Chemdataextractor: a toolkit for automated extraction of chemical information from the scientific literature. *Journal of chemical information and modeling*, 56(10):1894–1904, 2016.

[30] Daniel C Elton, Dhruv Turakhia, Nischal Reddy, Zois Boukouvalas, Mark D Fuge, Ruth M Doherty, and Peter W Chung. Using natural language processing techniques to extract information on the properties and functionalities of energetic materials from large text corpora. *arXiv preprint arXiv:1903.00415*, 2019.

[31] Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, et al. The chembl database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic acids research*, 52(D1):D1180–D1192, 2024.

[32] Kamal Choudhary and Mathew L Kelley. Chemnlp: a natural language-processing-based library for materials chemistry text data. *The Journal of Physical Chemistry C*, 127(35):17545–17555, 2023.

[33] Michiko Yoshitake, Yuta Suzuki, Ryo Igarashi, Yoshitaka Ushiku, and Keisuke Nagato. Materialbench: Evaluating college-level materials science problem-solving abilities of large language models, 2024.

[34] Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Benedict Emoekabu, Aswanth Krishnan, Mara Wilhelmi, Macjonathan Okereke, Juliane Eberhardt, Amir Mohammad Elahi, Maximilian Greiner, Caroline T. Holick, Tanya Gupta, Mehrdad Asgari, Christina Glaubitz, Lea C. Klepsch, Yannik Köster, Jakob Meyer, Santiago Miret, Tim Hoffmann, Fabian Alexander Kreth, Michael Ringleb, Nicole Roesner, Ulrich S. Schubert, Leanne M. Stafast, Dinga Wonanke, Michael Pieler, Philippe Schwaller, and Kevin Maik Jablonka. Are large language models superhuman chemists?, 2024.

[35] Taicheng Guo, Kehan Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. What indeed can gpt models do in chemistry? a comprehensive benchmark on eight tasks, 2023.

[36] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1):D1373–D1380, 2023.

[37] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

[38] Maria Sorokina, Peter Merseburger, Kohulan Rajan, Mehmet Aziz Yirik, and Christoph Steinbeck. Coconut online: collection of open natural products database. *Journal of Cheminformatics*, 13(1):2, 2021.

[39] Eliseu Pereira. Msds-opp: Operator procedures prediction in material safety data sheets. In *15th Doctoral Symposium*, page 42, 2020.

[40] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420, 2007.

[41] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[42] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[43] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759, 2009.

[44] Tanishq Gupta, Mohd Zaki, NM Anoop Krishnan, and Mausam. Matscibert: A materials domain language model for text mining and information extraction. *npj Computational Materials*, 8(1):102, 2022.

[45] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. PubChem 2023 update. *Nucleic Acids Research*, 51(D1):D1373–D1380, 10 2022.

[46] Eliseu Pereira. Msds-opp: Operator procedures prediction in material safety data sheets. 02 2020.

[47] Yonghui Wu. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[48] T Kudo. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[49] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[50] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[51] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[52] Liu Zheng and Shao Yingxia. Retromae: Pre-training retrieval-oriented transformers via masked auto-encoder. *arXiv: 2205.12035*, 2022.

[53] Zheng Liu, Shitao Xiao, Yingxia Shao, and Zhao Cao. Retromae-2: Duplex masked auto-encoder for pre-training retrieval-oriented language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2635–2648, 2023.

[54] Victor Fung, Jiaxin Zhang, Eric Juarez, and Bobby G Sumpter. Benchmarking graph neural networks for materials chemistry. *npj Computational Materials*, 7(1):84, 2021.

[55] Sterling G. Baird, Ramsey Issa, and Taylor D. Sparks. Materials science optimization benchmark dataset for multi-objective, multi-fidelity optimization of hard-sphere packing simulations. *Data in Brief*, 50:109487, 2023.

Supplementary Material

S1   Data Sources

PubChem [45] is a free, publicly accessible database of chemical molecules and their information including names, detailed descriptions, chemical formulas and properties, SMILES strings and 3D structures. We have employed PubChem to create pair classification and bitext mining datasets. One of our usages is to match SMILES strings (Isomeric or Canonical) with their corresponding entity titles and descriptions. The other usage is to match entity descriptions from different sources or their AI paraphrases to form pair classification datasets.

Wikipedia, with its vast array of articles, offers a comprehensive source of general and scientific knowledge, including chemistry-related content. For our specific use case, we extracted a chemistry-related subset of articles from the English Wikipedia and focused on tasks tailored to classification and clustering. Our contribution lies in creating different datasets with varying levels of difficulty and different numbers of classes, which were labeled by domain experts (chemists).

BeIR (Benchmarking Information Retrieval) [37] includes datasets like HotpotQA [42] and Natural Questions (NQ) [41] that focus on complex question-answering tasks, ideal for evaluating retrieval-based models. For our purposes, a chemistry-filtered subset of HotpotQA and NQ are leveraged for information retrieval tasks where the goal is to retrieve relevant text passages or documents in response to a query.

CoconutDB [38] is a database of natural products, which provides comprehensive information on molecular structures and properties. This database is particularly valuable for tasks involving the analysis of natural compounds. In our approach, we focused on bitext mining and pair classification between compound formulas and their corresponding SMILES representations.

Safety Data Sheets utilized in this study was sourced from Kaggle [46], which compiled over 200,000 Safety Data Sheets (SDS) through web scraping. Following the collection phase, a portion of the data was thoroughly cleansed and annotated to enhance quality and relevance. Our contribution lies in the creation of two specific label sets from the SDS data: $Gloves\_Required$ and $Eyes\_Protection\_Required$. Notably, the majority of SDS documents indicate whether protective gloves or eye protection are necessary. To convert this unstructured text into a structured format, where the requirement for gloves or eye protection is represented as a Boolean variable (required/not required), we developed an approach that combines large language models (LLMs) with regular expression techniques. This method enabled the efficient and accurate extraction of the relevant information from the raw text.

S2   Text Embedding Models

In this study, we evaluated several models on our chemical text embedding benchmark. These models are categorized into open-source and proprietary models based on their availability. Table S1 summarizes details about all the models that have been evaluated in this work.

S2.1   Open-source Models

BERT [4] introduced a groundbreaking approach to NLP by using the "masked language model" (MLM), which allows deep bidirectional context understanding by predicting masked tokens based on surrounding context. Additionally, BERT's "next sentence prediction" task enhances its ability to pre-train text-pair representations, enabling more accurate and versatile language models.

RoBERTa [5] is a replication and improvement of BERT, addressing several limitations in BERT's pretraining. RoBERTa introduces key modifications, including training for longer periods with larger batches, removing the next sentence prediction objective, training on longer sequences, and dynamically changing the masking pattern during training. Additionally, RoBERTa is trained on a significantly larger dataset, including the newly collected CC-NEWS corpus, which enhances its performance. These improvements allow RoBERTa to fully utilize the data, resulting in state-of-the-art performance on several GLUE tasks and matching top results on SQuAD and RACE.

13

SciBERT [6] builds upon the architecture of BERT, leveraging the same multilayer bidirectional Transformer model but tailored for scientific text. While BERT uses WordPiece tokenization [47] on general domain corpora, SciBERT constructs a specialized vocabulary (SCIVOCAB) from scientific papers using SentencePiece [48], resulting in a significant 42% difference in token overlap with BERT's BASEVOCAB. SciBERT is trained on 1.14 million papers from Semantic Scholar, covering computer science and biomedical domains. The resulting model excels in scientific NLP tasks, benefiting from the specialized corpus and vocabulary.

E5 [9], short for **E**mb**E**ddings from bidir**E**ctional **E**ncoder r**E**presentations, is a general-purpose text embedding model designed to generate high-quality, single-vector representations for a wide range of tasks in both zero-shot and fine-tuned settings. Unlike models that rely on limited labeled data or low-quality synthetic text pairs, E5 is contrastively trained on CCPairs, a curated web-scale dataset that incorporates diverse data sources such as CommunityQA, Common Crawl, and Scientific papers. To maintain high data quality, CCPairs undergoes rigorous filtering using consistency-based methods. E5 uses a two-stage contrastive learning approach, the first stage is done with unlabeled data and in-batch negatives with a large batch size, followed by a second stage of contrastive fine-tuning on a smaller labeled dataset and hard negative mining.

Nomic AI [10] proposed open-source embedding models with a focus on increased sequence length, efficiency, and accuracy for text embedding tasks (*nomic-embed-text* v1 and v1.5). It started the initial training on an improved version of BERT, called *Nomic-BERT* with substituting absolute positional embeddings for rotary positional embeddings [49], using SwiGLU activation instead of GeLU [50], and using FlashAttention [51]. They trained the base model with a maximum sequence length of 2048 and took advantage of Dynamic NTK interpolation to scale the sequence length to 8192 for inference. Subsequently, in addition to MLM, similar to E5, Nomic employed unsupervised contrastive pre-training and supervised contrastive fine-tuning to further boost its performance in embedding tasks.

BGE [11] leverages the RetroMAE [52, 53] framework for an efficient and effective pre-training phase, which involves recovering clean text from polluted text embeddings using a lightweight decoder. Following this, BGE undergoes a two-stage training process: contrastive learning with large batch sizes and in-batch negative sampling, followed by task-specific fine-tuning. The final phase incorporates instruction-based fine-tuning, where verbal prompts guide the model to better accommodate a variety of tasks. This multi-stage training pipeline ensures that BGE not only excels in general-purpose text embedding but also in specialized applications.

The M3-embedding [13] introduces several novel techniques to further enhance text embedding performance. One of the key innovations is self-knowledge distillation, which combines multiple outputs from different retrieval modes as a reward signal to boost the performance of single modes, particularly for sparse retrieval and multi-vector (ColBERT) retrieval. Additionally, M3 embedding improves efficiency when fine-tuning on long text through an efficient batching strategy. This small-batch approach is both simple and effective, and it can be applied to fine-tune large embedding models. Furthermore, the MCLS (Multiple CLS) method offers a straightforward way to improve performance on long texts without the need for fine-tuning, making it particularly useful for scenarios where resources for fine-tuning are limited.

S2.2   Proprietary Models

The following proprietary models were used in our benchmarks: OpenAI's *text-embedding-ada-002*, *text-embedding-3-large*, and *text-embedding-3-small*; Amazon's *amazon.titan-embed-text-v1* and *amazon.titan-embed-text-v2:0*; and Cohere's *cohere.embed-english-v3* and *cohere.embed-multilingual-v3*, accessed through Amazon Bedrock.

S3   More Details on Chemistry Benchmarks

MaterialBENCH [33] provided a dataset of problem-answer pairs for materials science and analyzing LLMs like ChatGPT and Bard on both free-response and multiple-choice questions for evaluating generative LLMs. ChemBench [34] introduced a framework to evaluate LLMs' chemical knowledge and reasoning abilities across diverse subfields, finding that while the best models often outperform human chemists, they struggle with some reasoning tasks and exhibit overconfidence in their pre-

Table S1: This table summarizes the embedding models, highlighting each model's name, Hugging-Face model or proprietary ID, model size on disk, number of parameters, the maximum context length, and the default embedding dimension. Models are categorized into open-source and proprietary sections for easier distinction.

	Model Name	HuggingFace Model / Model ID (Proprietary)	Model Size	# Parameters	Context length	Embedding size
	Open-Source Models					
1	BERT	google-bert/bert-base-uncased	440 MB	109.4 M	512	768
2	SciBERT	allenai/scibert_scivocab_uncased	442 MB	109.9 M	512	768
3	MatSciBERT	m3rg-iitd/matscibert	440 MB	109.9 M	512	768
4	Chemical BERT	recobo/chemical-bert-uncased	440 MB	109.9 M	512	768
5	Nomic BERT	nomic-ai/nomic-bert-2048	549 MB	136.7 M	2048	768
6	Nomic Embedding v1	nomic-ai/nomic-embed-text-v1	547 MB	136.7 M	8192	768
7	Nomic Embedding v1.5	nomic-ai/nomic-embed-text-v1.5	547 MB	136.7 M	8192	768
8	SBERT - all Mini LM L6.v2	sentence-transformers/all-MiniLM-L6-v2	90.9 MB	22.7 M	512	384
9	SBERT - all Mini LM L12.v2	sentence-transformers/all-MiniLM-L12-v2	133 MB	33.3 M	512	384
10	SBERT - all MPNET-base.v2	sentence-transformers/all-mpnet-base-v2	438 MB	109.4 M	514	768
11	SBERT - multi-qa-mpnet-base.v1	sentence-transformers/multi-qa-mpnet-base-dot-v1	438 MB	109.4 M	512	768
12	E5 - small	intfloat/e5-small	133 MB	33.3 M	512	384
13	E5 - base	intfloat/e5-base	438 MB	109.4 M	512	768
14	E5 - large	intfloat/e5-large	1.34 GB	335.1 M	512	1024
15	E5 - small v2	intfloat/e5-small-v2	133 MB	33.6 M	512	384
16	E5 - base v2	intfloat/e5-base-v2	438 MB	109.4 M	512	768
17	E5 - large v2	intfloat/e5-large-v2	1.34 GB	335.1 M	512	1024
18	E5 - Multilingual small	intfloat/multilingual-e5-small	471 MB	117.6 M	512	384
19	E5 - Multilingual base	intfloat/multilingual-e5-base	1.11 GB	278 M	514	768
20	E5 - Multilingual large	intfloat/multilingual-e5-large	2.24 GB	559.8 M	514	1024
21	BGE - small en	BAAI/bge-small-en	133 MB	33.3 M	512	384
22	BGE - base en	BAAI/bge-base-en	438 MB	109.4 M	512	768
23	BGE - large en	BAAI/bge-large-en	1.34 GB	335.1 M	512	1024
24	BGE - small en v1.5	BAAI/bge-small-en-v1.5	133 MB	33.3 M	512	384
25	BGE - base en v1.5	BAAI/bge-base-en-v1.5	438 MB	109.4 M	512	768
26	BGE - large en v1.5	BAAI/bge-large-en-v1.5	1.34 GB	335.1 M	512	1024
27	BGE - Multilingual - M3	BAAI/bge-m3	2.27 GB	576.7 M	8192	1024
	Proprietary Models					
28	OpenAI - Text embedding 3 - small	text-embedding-3-small	N/A	N/A	8191	1536
29	OpenAI - Text embedding 3 - large	text-embedding-3-large	N/A	N/A	8191	3072
30	OpenAI - Text embedding - Ada - 02	text-embedding-ada-002	N/A	N/A	8191	1536
31	Amazon - Titan Text Embedding v2	amazon.titan-embed-text-v2:0	N/A	N/A	8191	1536
32	Amazon - Titan Embedding G1 Text	amazon.titan-embed-text-v1	N/A	N/A	8191	1536
33	Cohere - Embed English V3	cohere.embed-english-v3	N/A	N/A	512	1024
34	Cohere - Embed Multilingual V3	cohere.embed-multilingual-v3	N/A	N/A	512	1024

dictions. ChemLLMBench [35] evaluated LLMs across eight chemistry-related tasks, revealing that GPT-4 performs best but highlights various limitations and the impact of in-context learning. MatDeepLearn [54] benchmarks GNNs for materials chemistry applications, identifying strengths in handling compositionally diverse datasets but also noting the high data requirements and limitations. Additionally, benchmarks such as those involving simulation of hyperparameter combinations in Materials Science Optimization Benchmark [55] aim to provide efficient and accurate models for optimization tasks, emphasizing the importance of surrogate models to reduce computational overhead while maintaining realistic task complexity. Hence, a benchmark with the capability of revealing the quality of embedding models in the representation of domain-specific chemical materials was missing in the field; helping researchers to find venues for further improvement of models, and industrial users to pick the most capable models to be used in their relevant tasks.

S4 Supplementary Methods

S4.1 Kernel Density Estimation

Kernel Density Estimation (KDE) is a non-parametric method used to estimate the probability density function of a random variable. Unlike parametric methods, KDE does not assume an underlying distribution for the data. Instead, it smooths the data points by placing a kernel, typically a Gaussian, at each data point. The sum of these kernels provides an estimate of the overall distribution. The KDE is controlled by a bandwidth parameter, which determines the width of the kernels and influences the smoothness of the resulting density estimate. A smaller bandwidth

produces a more sensitive estimate that captures more detail in the data, while a larger bandwidth results in a smoother, more generalized estimate. KDE is widely used in data visualization, such as for plotting smoothed histograms, and is applicable in fields like finance, biology, and machine learning, where understanding data distributions is critical.

S5   Supplementary Results

S5.1   Processing Time

Table S2 provides the details of processing time for all tasks on a V100 GPU machine.

Table S2: Average time (seconds) to run a benchmark for each task in each category on an Nvidia V100 32GB GPU instance.

	Classification	Bitext Mining	Retrieval	Clustering	Pair Classification
BERT	13.78±3.47	23.24±1.92	56.6±0.69	5.92±0.57	6.94±0.84
SciBERT	11.37±2.57	22.42±1.8	54.19±0.63	5.83±0.54	4.76±0.39
MatSciBERT	11.05±2.56	22.29±1.74	54.37±0.61	5.81±0.54	4.92±0.42
Chemical BERT	11.47±2.59	22.25±1.75	54.73±0.57	5.85±0.54	4.92±0.42
Nomic BERT	15.14±3.62	29.26±2.24	76.75±0.91	8.16±0.78	6.97±0.64
Nomic Embedding v1	23.13±5.03	31.05±2.4	79.45±0.88	12.01±1.24	5.91±0.41
Nomic Embedding v1.5	22.82±4.93	28.39±2.12	79.66±0.91	12.09±1.24	5.23±0.35
SBERT - all Mini LM L6.v2	2.36±0.52	9.01±0.89	12.83±0.3	1.13±0.11	1.73±0.14
SBERT - all Mini LM L12.v2	2.82±0.57	11.73±1.08	16.26±0.39	1.17±0.1	1.99±0.09
SBERT - all MPNET-base.v2	11.36±2.73	24.49±1.87	61.51±0.76	6.26±0.6	4.43±0.29
SBERT - multi-qa-mpnet-base.v1	13.29±3.09	24.06±1.92	62.06±0.67	7.13±0.69	4.42±0.3
E5 - small	4.98±1.06	12.55±1.11	21.79±0.29	2.36±0.22	2.54±0.2
E5 - base	11.24±2.67	23.93±1.97	58.84±0.75	6.28±0.6	5.28±0.46
E5 - large	37.37±9.5	62.4±4.78	191.41±2.06	20.46±1.99	14.83±1.43
E5 - small v2	5.34±1.08	12.63±1.12	22.15±0.25	2.45±0.23	2.39±0.21
E5 - base v2	11.21±2.66	24.27±2.02	59.45±0.73	6.34±0.6	4.83±0.49
E5 - large v2	36.87±9.28	64.27±4.96	193.9±2.14	20.54±1.97	14.49±1.47
E5 - Multilingual small	5.2±1.11	11.96±1.06	21.68±0.28	2.37±0.23	2.29±0.2
E5 - Multilingual base	12.51±2.99	23.96±1.97	62.13±0.65	6.82±0.67	4.74±0.48
E5 - Multilingual large	40.26±10.31	60.97±4.51	209.69±0.52	22.01±2.18	13.8±1.43
BGE - small en	5.23±1.05	12.46±1.1	21.64±0.29	2.32±0.22	2.82±0.19
BGE - base en	11.14±2.64	23.99±1.98	58.64±0.72	6.29±0.6	5.32±0.48
BGE - large en	37.04±9.33	62.27±4.79	191.56±2.06	20.44±1.97	14.89±1.44
BGE - small en v1.5	5.28±1.05	12.37±1.07	21.83±0.25	2.39±0.23	2.68±0.19
BGE - base en v1.5	11.14±2.63	23.82±1.99	59.08±0.8	6.27±0.59	5.27±0.46
BGE - large en v1.5	36.57±9.12	62.24±4.8	191.63±2.14	20.41±1.97	14.85±1.43
BGE - Multilingual - M3	1139.9±251.82	707.86±48.87	3031.81±22.43	640.67±75.54	31.82±8.61
OpenAI - Text embedding 3 - small	37.17±6.89	372.97±36.14	518.72±12.57	27.74±2.46	63.49±2.91
OpenAI - Text embedding 3 - large	62.18±11.8	730.16±70.34	1006.01±27.68	49.39±4.65	123.33±5.89
OpenAI - Text embedding - Ada - 02	35.57±6.77	372.55±36.18	518.73±12.83	30.77±1.88	64.41±2.94
Amazon - Titan Text Embedding v2	128.01±35.05	1178.06±99.02	1595.24±34.49	84.65±7.8	244.12±3.41
Amazon - Titan Embedding G1 Text	142.23±37.78	1174.83±97.29	1627.31±40.39	89.03±8.38	243.45±3.53
Cohere - Embed English V3	21.21±5.64	83.08±5.98	134.29±2.48	13.27±1.25	16.65±0.89
Cohere - Embed Multilingual V3	22.32±6.07	80.27±5.86	138.74±2.51	14.08±1.3	18.07±1.29

S5.2 Ranking of models

Detailed ranking of models on each category of tasks is provided in Table S3. The ranking is calculated based on average performance over all tasks in each category defined on them.

Table S3: Summary of models rank

	Classification	Bitext Mining	Retrieval	Clustering	Pair Classification	RRF_Score(k=10)
Nomic BERT	34	33	34	27	34	0.118
Chemical BERT	33	30	32	33	31	0.120
MatSciBERT	32	31	33	28	32	0.122
BERT	30	34	30	29	33	0.122
SciBERT	31	32	31	31	30	0.122
BGE - small en	12	27	26	25	22	0.160
E5 - small v2	23	25	8	30	29	0.165
E5 - small	25	21	9	34	24	0.166
BGE - Multilingual - M3	21	26	10	15	28	0.176
E5 - base v2	22	18	11	19	25	0.178
BGE - small en v1.5	9	23	18	26	20	0.180
SBERT - multi-qa-mpnet-base.v1	28	29	24	17	5	0.185
BGE - base en	16	13	22	16	19	0.186
E5 - Multilingual large	27	7	14	24	23	0.187
BGE - large en	10	19	29	13	17	0.191
E5 - base	20	14	12	22	15	0.192
E5 - Multilingual base	26	11	13	10	27	0.196
SBERT - all Mini LM L12.v2	18	22	23	21	4	0.201
E5 - Multilingual small	29	16	1	32	26	0.207
E5 - large v2	24	12	3	23	21	0.214
BGE - base en v1.5	15	17	7	11	18	0.219
BGE - large en v1.5	6	15	15	18	12	0.224
Amazon - Titan Text Embedding v2	17	8	19	8	14	0.224
SBERT - all Mini LM L6.v2	8	20	20	20	3	0.232
SBERT - all MPNET-base.v2	7	28	25	6	6	0.239
OpenAI - Text embedding 3 - small	5	5	17	9	10	0.273
Cohere - Embed English V3	2	24	28	2	8	0.278
OpenAI - Text embedding - Ada - 02	11	2	16	4	16	0.279
Cohere - Embed Multilingual V3	4	9	27	3	9	0.281
Nomic Embedding v1	19	10	4	12	2	0.285
Amazon - Titan Embedding G1 Text	1	3	21	14	13	0.285
E5 - large	14	4	6	5	11	0.290
Nomic Embedding v1.5	13	6	2	7	1	0.339
OpenAI - Text embedding 3 - large	3	1	5	1	7	0.384

S5.3 Comparison of MTEB and ChemTEB

Figure S1 reflects the performance of each model in each category of tasks on both benchmarks. In three out of four categories of tasks, the BERT model provided the weakest performance in both benchmarks.
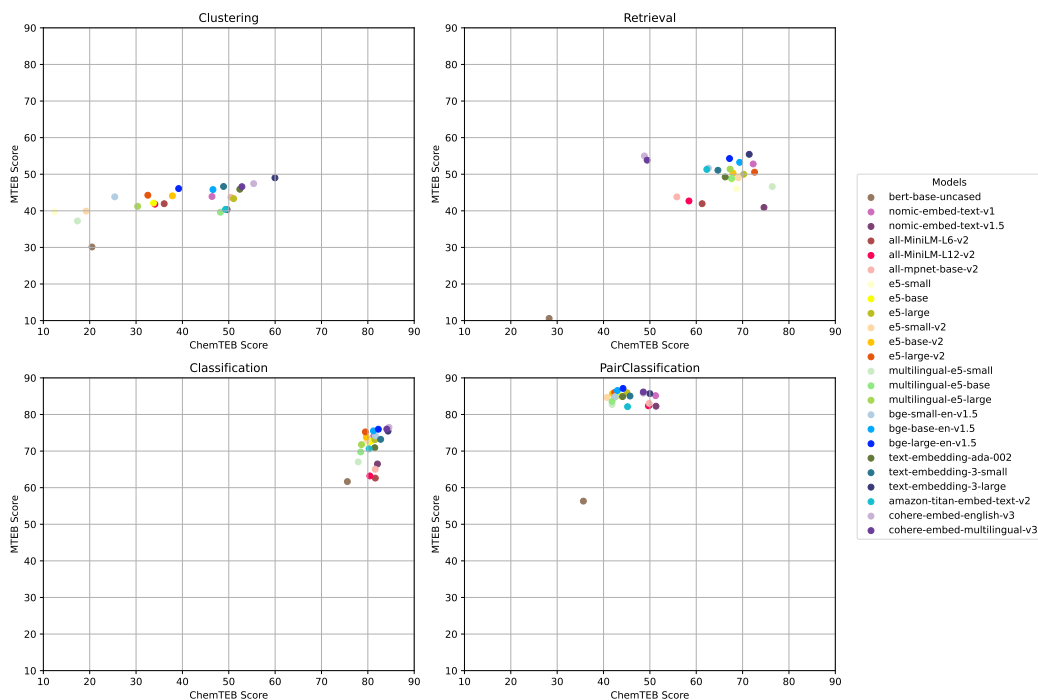

Figure S1: Comparison of model performance on ChemTEB and MTEB benchmarks across different tasks. Each point represents a model from the intersection of those tested and those on the MTEB leaderboard as of the date. The figure highlights variations in task difficulty and domain specificity.

S5.4 Correlation between models performances and tasks

Figures S2 and S3 illustrate the correlation matrix for the datasets and models, respectively, with colors representing the strength of the correlations. In figure S2, We can observe that in tasks such as classification, bitext mining, and retrieval, the datasets in a task are correlated except for the SDS datasets in the classification. In the clustering, and pair classification task, however, this trend is not very obvious. Especially, in the pair classification task, some of the datasets have negative correlation.
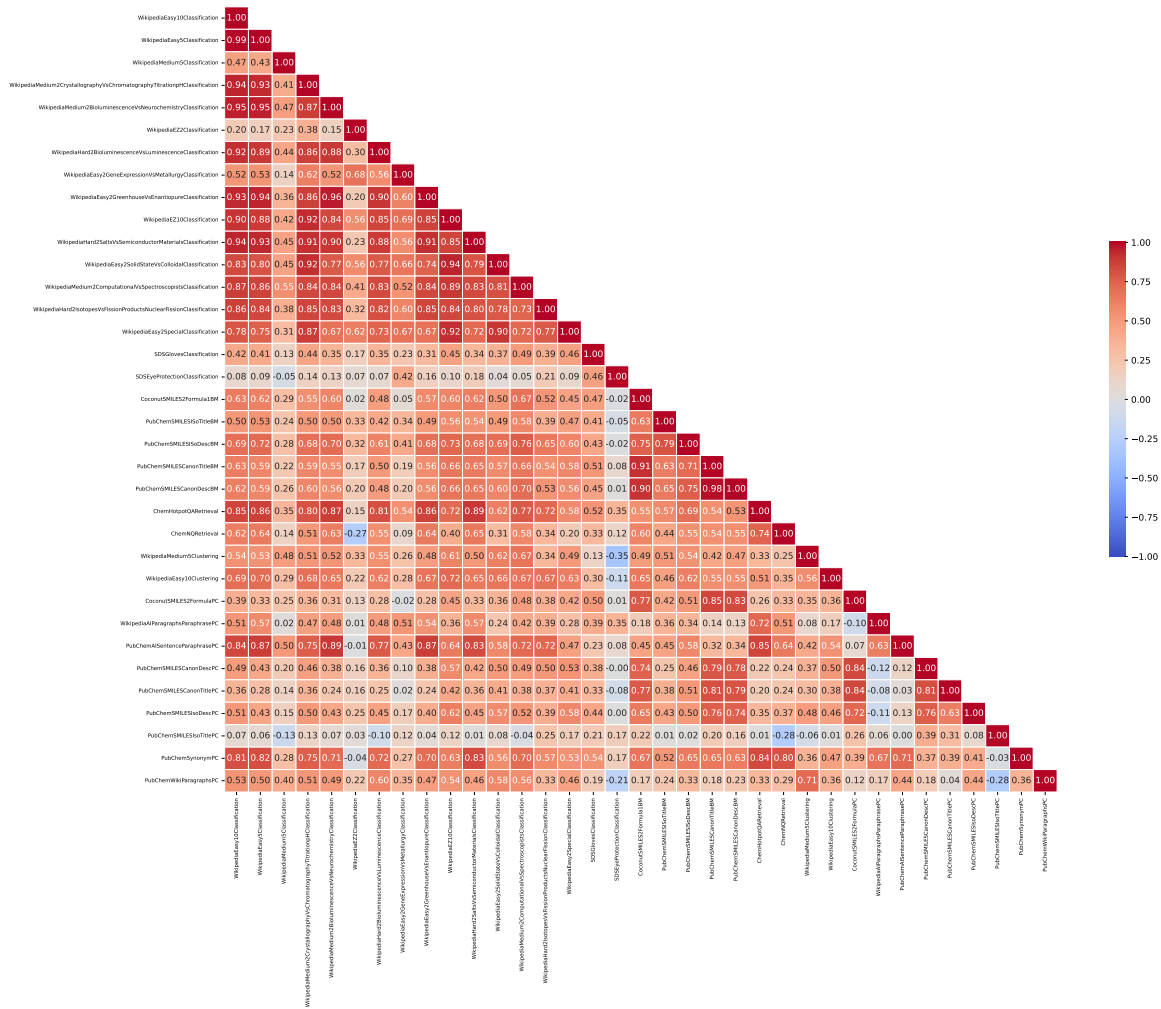

Figure S2: Correlation Matrix across datasets. Each row and column represents a separate dataset tested in the ChemTEB benchmark. The values and associated color reflect the correlation between the performance of different models on each pair of these datasets.

On the other hand, in figure S3, we can see models categorized into two groups. Group one, the BERT-based models, and group two, the other models. The biggest difference between these two groups is the contrastive learning being done after the pre-training, which, putting that aside the fact that BERT-based models have almost the worst performance in every single category of tasks, shows the importance of contrastive learning. Within the second group, some model families have a stronger correlation between their performance, which may reflect closer architecture and/or pre-training tasks. For example, the Nomic embedding family has the highest correlation with the SBERT family. Cohere also reflects a closer correlation to the SBERT family first, and then the Amazon models family in the second place.

Figure S3: Correlation Matrix over Models. Each row and column represents a separate Model tested in the ChemTEB benchmark. The values and associated color reflects the correlation between the performance of each pair of models over all tested datasets.