HuggingMolecules: an open-source library for transformer-based molecular property prediction

Piotr Gaiński[1], Łukasz Maziarka[1,2], Tomasz Danel[1,2],
Stanisław Jastrzębski[1,3]

[1]Jagiellonian University [2]Ardigen [3]Molecule.one

Abstract

Large-scale transformer-based methods are gaining popularity as a tool for predicting the properties of chemical compounds, which is of central importance to the drug discovery process. To accelerate their development and dissemination among the community, we are releasing **HuggingMolecules – an open-source library, with a simple and unified API, that provides implementation of several state-of-the-art transformers for molecular property prediction**. In addition, we add a comparison of these methods on several regression and classification datasets.

Code snippet

```python
from huggingmolecules import MatModel, MatFeaturizer

from experiments.src import TrainingModule,
                            get_data_loaders

from torch.nn import MSELoss
from torch.optim import Adam

from pytorch_lightning import Trainer
from pytorch_lightning.metrics import MeanSquaredError

# Build and load the pre-trained model
# and the appropriate featurizer:
model = MatModel.from_pretrained('mat_masking_20M')
featurizer = MatFeaturizer.from_pretrained(
                            'mat_masking_20M')

# Build the pytorch lightning training module:
pl_module = TrainingModule(model,
                    loss_fn=MSELoss(),
                    metric_cls=MeanSquaredError,
                    optimizer=Adam(model.parameters()))

# Build the data loader for the FreeSolv dataset:
train_dataloader,_,_ = get_data_loaders(featurizer,
                        batch_size=32,
                        task_name='ADME',
                        dataset_name=
                        'hydrationfreeenergy_freesolv')

# Build the pytorch lightning trainer and
# fine-tune the module on the train dataset:
trainer = Trainer(max_epochs=100)
trainer.fit(pl_module,
            train_dataloader=train_dataloader)

# Make the prediction for the batch of SMILES strings:
batch = featurizer(['C/C=C/C', '[C]=O'])
output = pl_module.model(batch)
```

References

[1] Łukasz Maziarka, Tomasz Danel, Sławomir Mucha, Krzysztof Rataj, Jacek Tabor, and Stanisław Jastrzębski. Molecule attention transformer. *arXiv preprint arXiv:2002.08264*, 2020.

[2] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33, 2020.

[3] Seyone Chithrananda, Gabe Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

[4] Benedek Fabian, Thomas Edlich, Héléna Gaspar, Marwin Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020.

[5] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

Models

Table 1: Models used in our benchmark.

Model name	Citation	Type	No. params
MAT	[1]	graph-based	42M
GROVER	[2]	graph-based	48M/107M
ChemBERTa	[3]	SMILES-based	83M
MolBert	[4]	SMILES-based	85M
D-MPNN	[5]	graph-based	355k

Datasets

Table 2: Datasets used in our benchmark.

Dataset	Category	Task type	Compounds	Metric	Split method	from TDC
FreeSolv	ADME	regression	642	RMSE	random	yes
Caco-2	ADME	regression	910	RMSE	random	yes
Clearance	ADME	regression	731	RMSE	random	yes
QM7	ADME	regression	6830	MAE	random	no
HIA	ADME	classification	578	ROC AUC	random	yes
Bioavailability	ADME	classification	640	ROC AUC	random	yes
PPBR	ADME	classification	765	ROC AUC	random	yes
BBBP	ADME	classification	2039	ROC AUC	scaffold	no
Tox21 (NR-AR)	ADME	classification	7256	ROC AUC	random	yes

Benchmark results

Table 3: Benchmark results for the regression tasks. As the metric we used MAE for QM7 and RMSE for the rest of datasets.

	FreeSolv	Caco-2	Clearance	QM7	Mean rank
MAT 200k	.913 ± .196	**.405 ± .030**	.649 ± .341	87.578 ± 15.37	5.25
MAT 2M	.898 ± .165	.471 ± .070	.655 ± .327	81.557 ± 5.08	6.75
MAT 20M	**.854 ± .197**	.432 ± .034	.640 ± .335	81.797 ± 4.17	5.0
GROVER Base	.917 ± .195	.419 ± .029	.629 ± .335	**62.27 ± 3.58**	3.25
GROVER Large	.950 ± .202	.414 ± .041	**.627 ± .340**	64.94 ± 3.62	**2.5**
ChemBERTa	1.218 ± .245	.430 ± .013	.647 ± .314	177.242 ± 1.81	8.0
MolBERT	1.027 ± .244	.483 ± .056	.633 ± .332	177.117 ± 1.79	8.0
D-MPNN	1.061 ± .168	.446 ± .064	.628 ± .339	74.83 ± 4.79	5.5
D-MPNN 2d	1.038 ± .235	.454 ± .049	.628 ± .336	77.91 ± 1.21	6.0
D-MPNN mc	.995 ± .136	.438 ± .053	**.627 ± .337**	75.58 ± 4.68	4.25

Table 4: Benchmark results for the classification tasks. We used ROC AUC as the metric.

	HIA	Bioavailability	PPBR	Tox21 (NR-AR)	BBBP	Mean rank
MAT 200k	**.943 ± .015**	.660 ± .052	.896 ± .027	.775 ± .035	.709 ± .022	5.8
MAT 2M	.941 ± .013	.712 ± .076	**.905 ± .019**	**.779 ± .056**	.713 ± .022	4.2
MAT 20M	.935 ± .017	.732 ± .082	.891 ± .019	**.779 ± .056**	**.735 ± .006**	**3.4**
GROVER Base	.931 ± .021	.750 ± .037	.901 ± .036	.750 ± .085	**.735 ± .006**	4.0
GROVER Large	.932 ± .023	**.747 ± .062**	.901 ± .033	.757 ± .057	.728 ± .005	4.2
ChemBERTa	.923 ± .032	.666 ± .041	.869 ± .032	**.779 ± .044**	.717 ± .009	7.0
MolBERT	.942 ± .011	.737 ± .085	.889 ± .039	.761 ± .058	.742 ± .020	4.6
D-MPNN	.924 ± .069	.724 ± .0644	.847 ± .052	.766 ± .040	.726 ± .008	7.0
D-MPNN 2d	.900 ± .094	.712 ± .067	.874 ± .030	.775 ± .041	.724 ± .006	6.8
D-MPNN mc	.924 ± .082	.740 ± .060	.869 ± .033	.772 ± .041	.722 ± .008	6.2

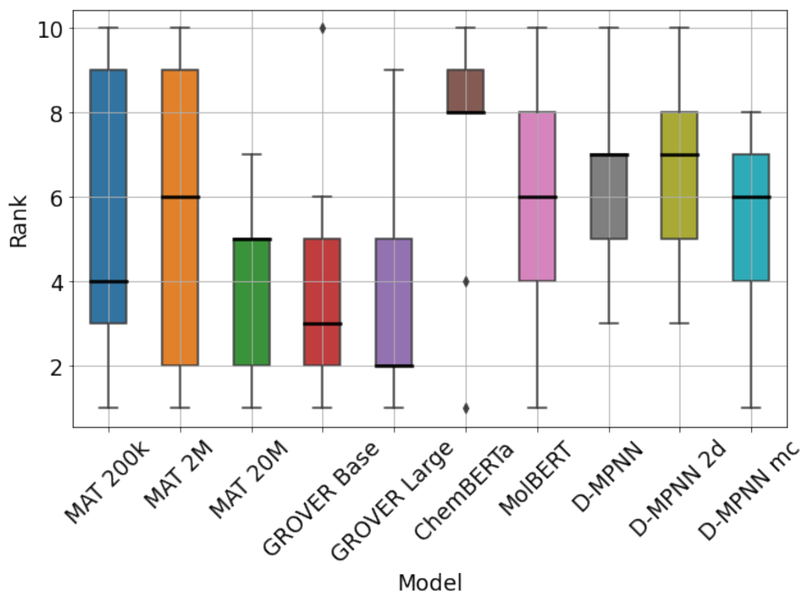

Figure 1: Rank plot for the datasets from our benchmark. We can see that the graph-base transformers outperforms these based on SMILES, moreover they beat D-MPNN, which is the non-transformer state-of-the-art in molecular property prediction tasks.