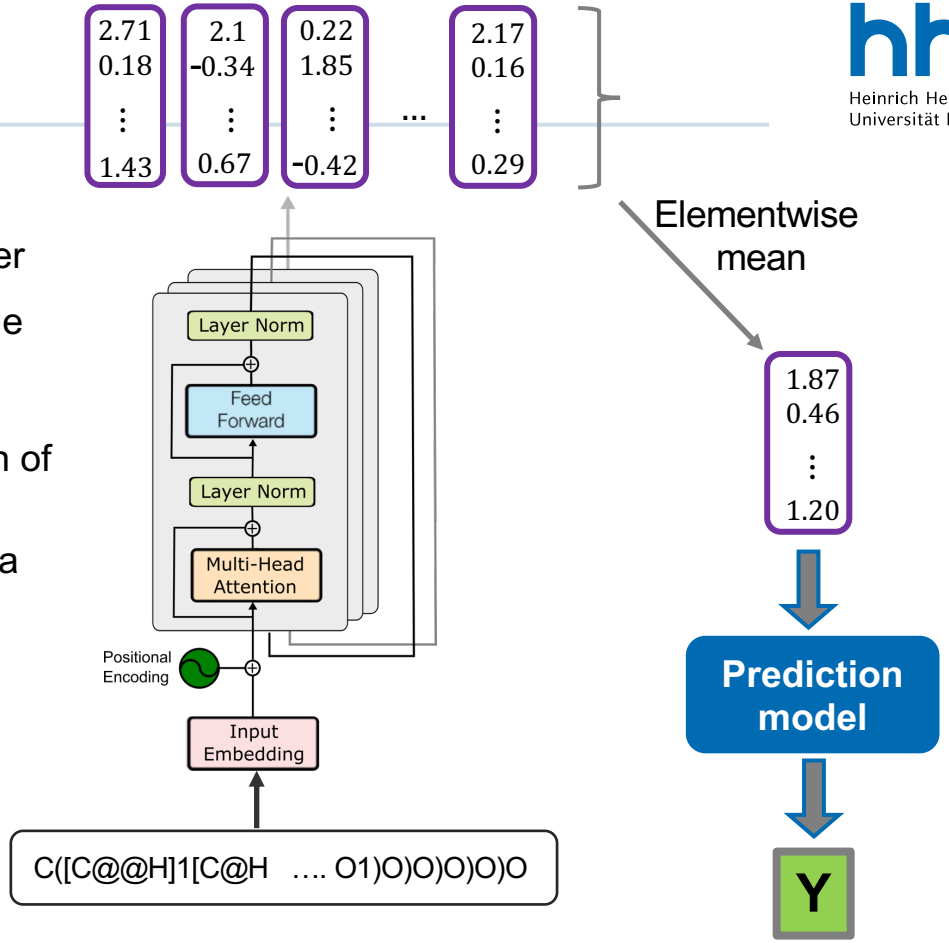
Heinrich Heine
Universität Düsseldorf

Transformer Networks for small molecules

Motivation

- We want to use SMILES strings as the input for a Transformer Network Encoder
- After training, we want to extract a single numerical vector representing the small molecule
  - For example, the elementwise mean of all token representations
- Use the resulting vector as the input of a machine learning prediction model



Tokenization of SMILES strings (1)

- Not as straightforward as for protein amino acid sequences
- Options for tokenization:
  - Each character is a separate token
  - Search for most common patterns
    - Byte Pair Encoding:

SMILES dataset:

C[C@H](N)C(=O)O
C/C=C/C(=O)O
C[C@H](O)[C@@H](O)C(=O)O
F[C@](Cl)(Br)I
…
….
C[C@H]([C@@H](C(=O)O)N)O
C[C@@H]1CN(C)C[C@H]1C
C[C@H](C)[C@H](C(=O)O)N

Initial set of tokens:

- C
- [
- @
- …
-]
- (
- N
-)

Additional tokens:

- [C
- [C@
- …

Special Tokens:

- <bos>
- <eos>
- <pad>
- <mask>
- …

Most common combination of existing tokens:

1. “[“ + “C” = “[C”
2. “[C“ + “@” = “[C@”
3. …

Tokenization of SMILES strings (2)

- Options for tokenization:
  - Based on SMILES rules

Special Tokens:
- <bos>
- <eos>
- <pad>
- <mask>
- …

Single Characters:
- (
-)
- /
- \
- -
- =
- #
- 1
- 2
- …
- 9
- C
- O
- N
- F
- S
- P
- I
- B
- Cl
- Br
- c
- o
- …

Complex Tokens:
- [C@H]
- [C@@H]
- [C@]
- [C@@]
- [N+]
- [O-]
- [S@]
- [S@@]
- [NH+]
- [Fe]
- …
- [N-]
- [Si]
- [n+]
- [2H]
- [nH]
- [Na+]
- [Cl-]
- [c-]
- [C-]
- …

Positional embeddings

- Options for positional embeddings:
  - Learned embeddings

$$\begin{pmatrix} p_{1,1} & \cdots & p_{1,d} \\ \vdots & \ddots & \vdots \\ p_{512,1} & \cdots & p_{512,d} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} p_{2,1} \\ p_{2,2} \\ \vdots \\ p_{2,d} \end{pmatrix}$$

  - Sinusoidal positional encodings

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right),$$

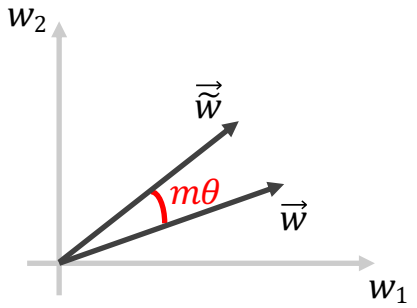
$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right)$$

  - Rotary Positional Embeddings (RoPE)

m position in input sequence

θ rotation constant

$$\vec{w} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \xrightarrow{\text{rotate by } m \cdot \theta} \vec{\tilde{w}}$$



Model training

- Masked Language Modelling (MLM):
  - Masking 15% of the tokens in each input string
  - Training the model to correctly identify the masked tokens
  - Problems:
    - In contrast to language/proteins it is much less restricted what token you can have at what positions

- Multi-task Regression (MTR)
  - Compute a set of 200 molecular properties for each compound in our training dataset
  - Properties can be calculated from SMILES strings using RDKit

MLM vs. MTR

Heinrich Heine
Universität Düsseldorf

	BACE *RMSE*	Clearance *RMSE*	Delaney *RMSE*	Lipo *RMSE*	BACE *ROC*	BBBP *ROC*	ClinTox *ROC*	SR-p53 *ROC*
ChemBERTa-2								
MLM-5M	1.451	54.601	0.946	0.986	0.793	0.701	0.341	0.762
MLM-10M	1.611	53.859	0.961	1.009	0.729	0.696	0.349	0.748
MLM-77M	1.509	52.754	1.025	0.987	0.735	0.698	0.239	0.749
MTR-5M	1.477	50.154	0.874	0.758	0.734	**0.742**	0.552	**0.834**
MTR-10M	1.417	48.934	**0.858**	**0.744**	0.783	0.733	0.601	0.827
MTR-77M	1.363	**48.515**	0.889	0.798	0.799	0.728	0.563	0.817

Table copied and modified from: https://arxiv.org/pdf/2209.01712

Popular small molecule Transformer

Heinrich Heine Universität Düsseldorf

	ChemBERTa-2[1]	Molformer[2]
Architecture	Encoder only	Encoder only
Model size	3.4M	~85M
Pre-training task	MTR	MLM
Training set size	77M	1.1B
Input	canonical SMILES	canonical SMILES
Pos. embedding	Learned	RoPE
Tokenization	Frequency-based method?	Based on SMILES rules

[1] https://arxiv.org/pdf/2209.01712, [2] https://arxiv.org/pdf/2106.09553