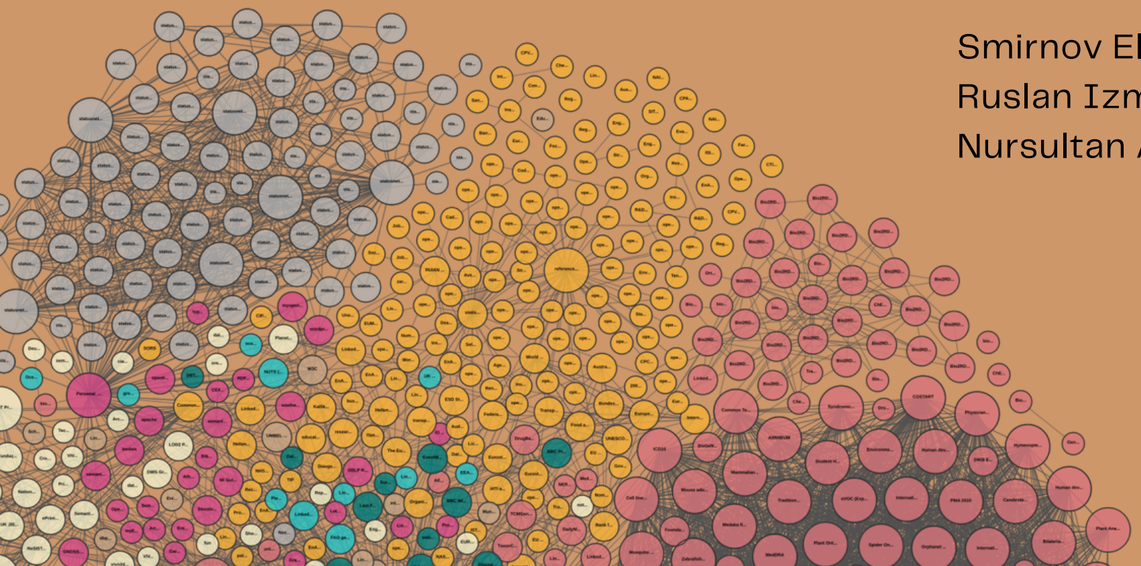
GRAPH TRANSFORMERS



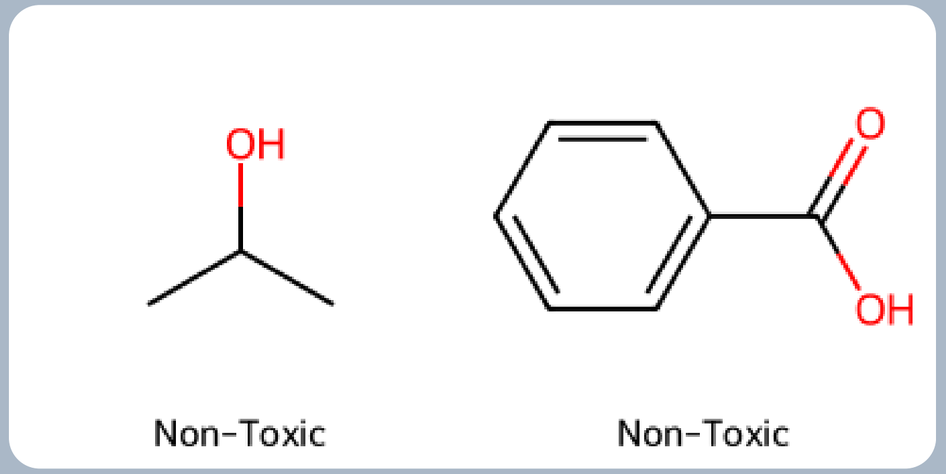
github.com/Hexy00123/F24-DKR-Project

Smirnov Elisei
Ruslan Izmailov
Nursultan Abdullaev

Problem Statement

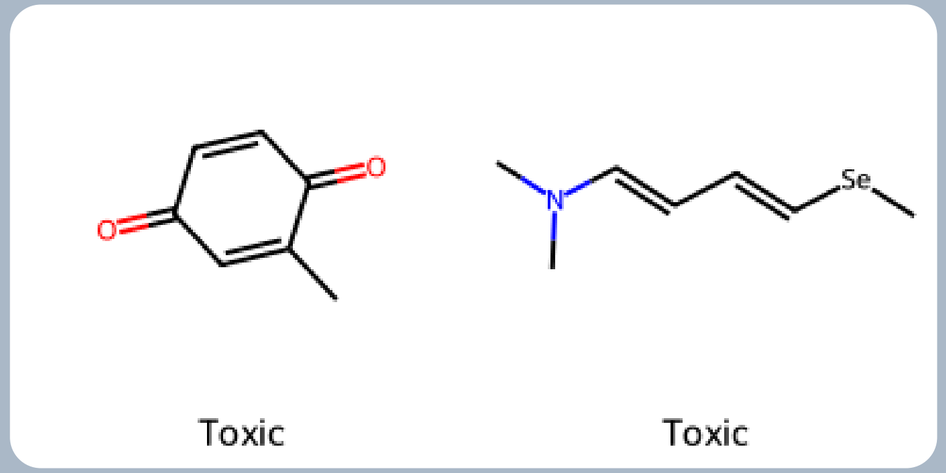
Challenge: Identifying toxic molecules in chemical datasets

Why it Matters: Essential for drug discovery and safety evaluation



isopropanol

benzoic acid



maleic anhydride

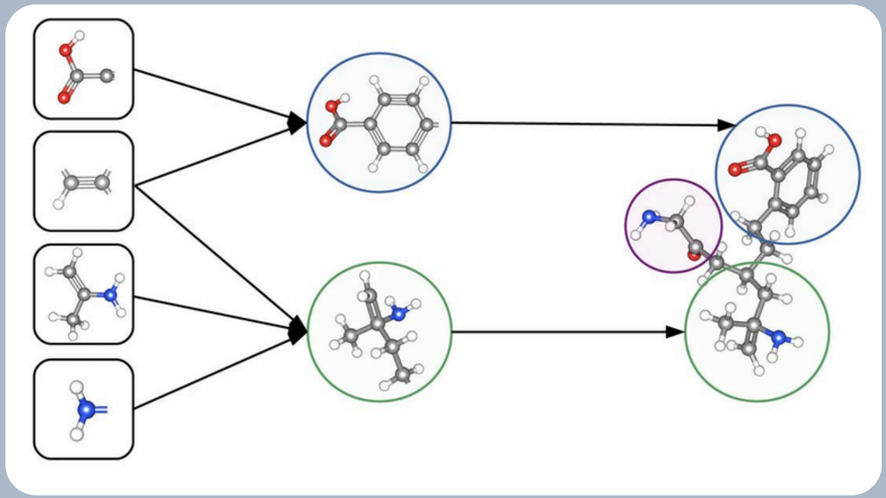
selenourea derivative

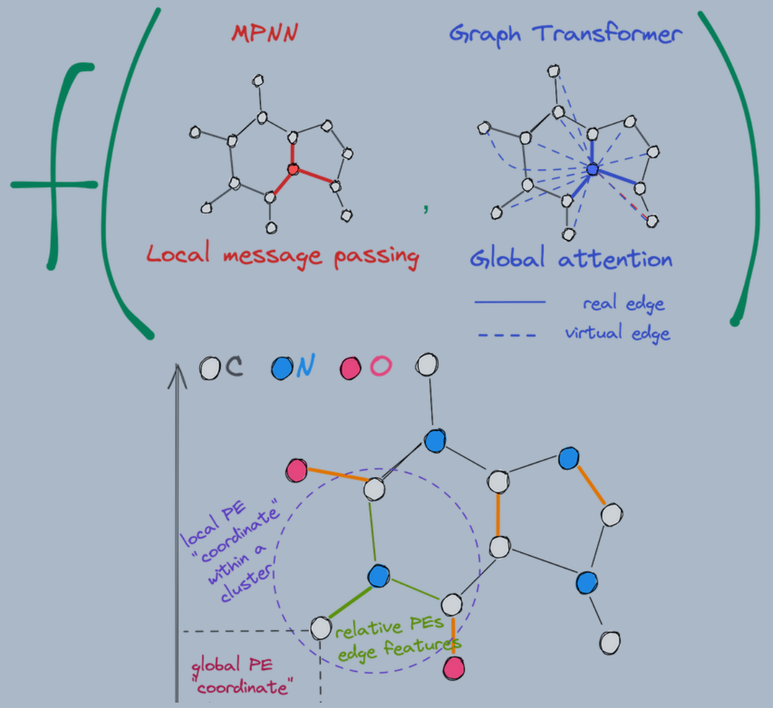
Methodology Overview

1) **Represent molecules as graphs** (atoms = nodes, bonds = edges) –> Tox21 dataset

2) **Extract features with Morgan fingerprints and adjacency matrices**

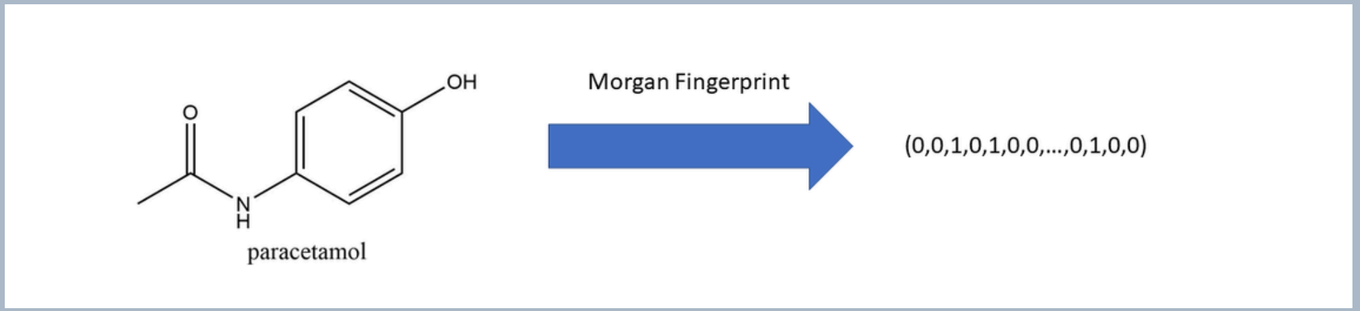
3) **Train a Graph Transformer model on toxicity prediction**





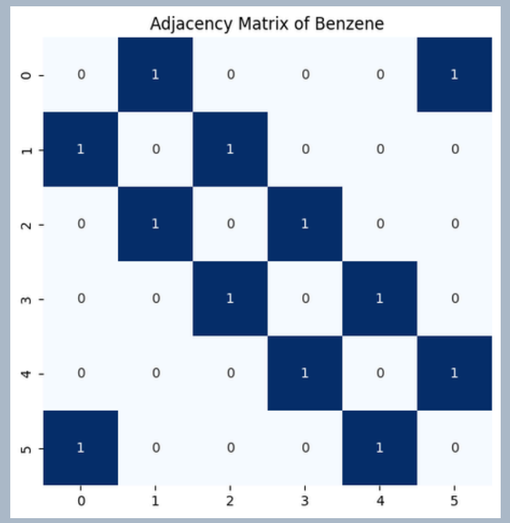
Extract features

1) Morgan fingerprints – are type of molecular descriptor used in computational chemistry.



If the feature is present, the bit is set to 1; otherwise, it is 0.

2) Adjacency Matrix



The matrix entry Adj[i][j] is 1 if there is a bond between atom i and atom j, and 0 otherwise.

Graph Transformer model

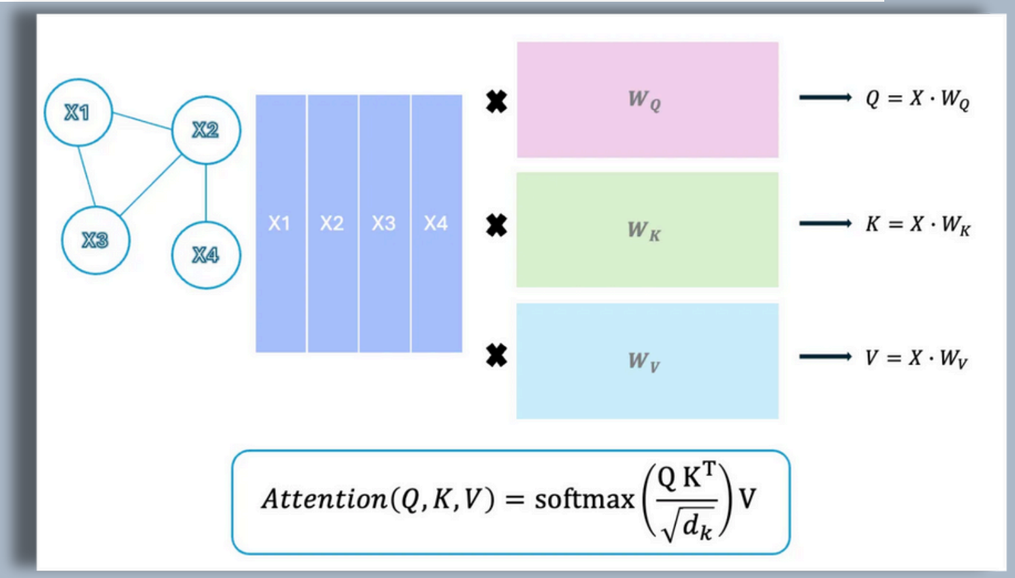
/ Code_Explained.md

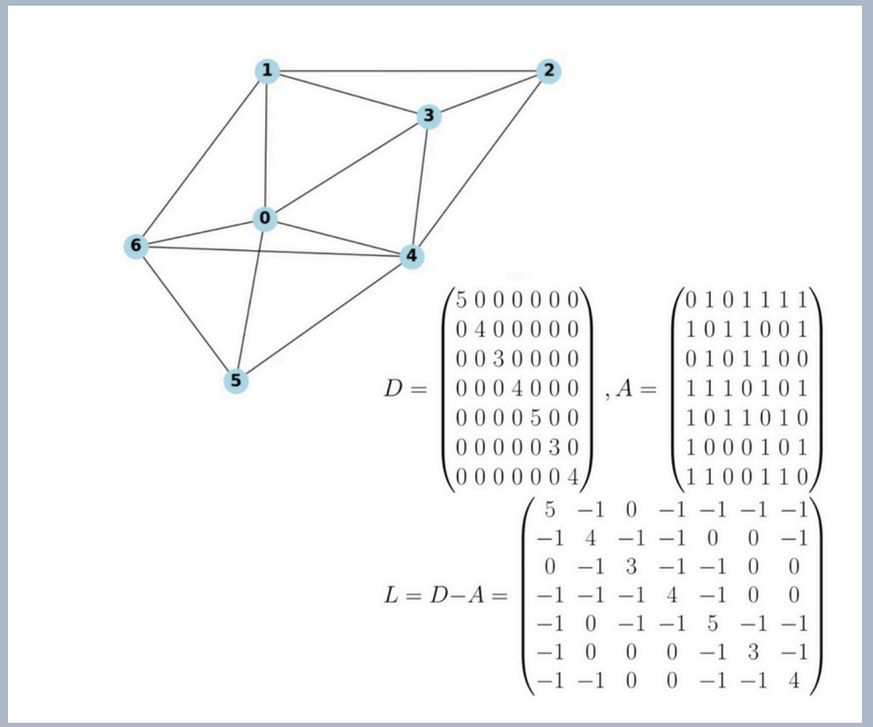
ne 996 lines (685 loc) · 44.4 KB Code 55% faster with GitHub Copilot

Graph Transformer Architecture Overview

The architecture of a Graph Transformer includes the following components:

1. **Input Graph**: The model takes an input graph with nodes x_1, x_2, x_3, x_4 and edges representing the relationships between nodes.
2. **Node Features**: Each node x_i in the graph is initially represented by a feature vector $X = [x_1, x_2, x_3, x_4]$.
3. **Linear Transformations**:
   - The node feature matrix X is multiplied by three learned weight matrices W_Q, W_K, W_V to produce the **Query (Q)**, **Key (K)**, and **Value (V)** matrices.
4. **Positional Encoding**:
   - Unlike NLP Transformers, where positional encodings capture the position of words in a sequence, **Graph Transformers use positional encodings to capture the structural information of the graph**, providing context about the node's position within the graph.





Positional Encoding of Nodes Using Laplacian Eigenvectors

Results

- Accurate toxicity prediction on Tox21 dataset is 70% with 2 epochs trained only!
- Graph Transformers excelled at capturing molecular interactions

```
Training: 100%|██████████| 735/735 [00:56<00:00, 13.08it/s]
Validating: 100%|██████████| 5873/5873 [00:20<00:00, 284.22it/s]
Validating: 100%|██████████| 1958/1958 [00:06<00:00, 289.37it/s]
Epoch 1/2: Loss: 0.6668, Train Acc: 0.6620, Test Acc: 0.6879
Training: 100%|██████████| 735/735 [00:57<00:00, 12.86it/s]
Validating: 100%|██████████| 5873/5873 [00:21<00:00, 273.15it/s]
Validating: 100%|██████████| 1958/1958 [00:06<00:00, 290.91it/s]
Epoch 2/2: Loss: 0.6346, Train Acc: 0.6874, Test Acc: 0.7074
```

```python
new_smiles = "O" # water
prediction = predict(model, feature_extractor, new_smiles, device=device)
print(f"The predicted toxicity of the molecule with SMILES {new_smiles} is: {prediction}")
```

```
The predicted toxicity of the molecule with SMILES O is: 0
```