Molecule Retrieval with Natural Language Queries

Sofiane Ezzehi[1] **and Bastien Le Chenadec**[1]

[1]École des Ponts ParisTech

Contribution Statement

1 Introduction

The goal of this challenge is to retrieve molecules from a database using natural language queries. Each sample in the dataset is constituted of a ChEBI description of a molecule, which is a text describing its structure and properties, and an undirected graph representing the molecule with embeddings for each node. The embeddings are pre-computed using the Mol2Vec algorithm [1]. Given a textual query, the goal is to retrieve the molecule that best matches the query. The evaluation metric is the label ranking average precision score (LRAP) which is equivalent to the mean reciprocal rank (MRR) in our case.

The challenging part of this task is to find a way to combine two very different modalities : texts and graphs. One way to achieve this is to use contrastive learning : one model encodes the text and the other encodes the graph. The two encoders are then trained to project similar samples close to each other in the embedding space. This approach has been shown to be effective in many tasks [2, 3].

2 Data

3 Method

In this section, we describe the different models we used to encode the text and the graph. The text encoder and graph encoders are two separate models that are trained jointly using contrastive learning, so that they share the same embedding space.

3.1 Graph Attention Networks

Graph Attention Networks (GAT) [4] have been shown to be effective in many tasks. Like other graph neural networks, GATs aggregate information from the neighbors of each node to compute its embedding. The main difference with other models is that GATs use an attention mechanism to weight the neighbors of each node. Specifically we used the improved version of GATs suggested in [5].

Let G be an undirected graph with N nodes denoted $[\![1, N]\!]$. Let d be the dimension of the node embeddings, and $h_1, \dots, h_N \in \mathbb{R}^d$ be the said embeddings. Let $W \in \mathbb{R}^{d' \times d}$ and $a \in \mathbb{R}^{2d'}$. The attention weights are :

$$e(h_i, h_j) = a^T \, \text{LeakyReLU}([Wh_i \| Wh_j]) \tag{1}$$

where $\|$ denotes the concatenation operator. The attention weights are normalized using the softmax operator :

$$\alpha_{ij} = \frac{\exp(e(h_i, h_j))}{\sum_{k \in \mathcal{N}_i} \exp(e(h_i, h_k))} \tag{2}$$

where $\mathcal{N}_i$ denotes the set of neighbors of node i in G. This mechanism clearly allows the batch processing of graphs with different sizes. The embedding of node i is then computed as :

$$h'_i = \text{LeakyReLU}\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W h_j\right) \tag{3}$$

In general we will use multi-head attention, with K heads, $a^{(1)}, \dots, a^{(K)} \in \mathbb{R}^{2d'/K}$ and $W^{(1)}, \dots, W^{(K)} \in \mathbb{R}^{d'/K \times d}$:

$$h'_i = \text{LeakyReLU}\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(k)} W^{(k)} h_j\right) \tag{4}$$

Furthermore, we will stack multiple GAT layers to obtain a deeper model. We may also apply a multi-layer perceptron to the embeddings of the last layer to obtain a more expressive representation.

3.2 DiffPool

[6]

3.3 Language modelling

We used a pretrained large language model (LLM) to encode the text. Specifically, we settled on `sentence-transformers/all-MiniLM-L6-v2` which is a distilled version of MiniLM [7] known for its efficiency and relatively small size. The sentence embeddings are obtained by averaging the embeddings of the tokens in the sentence, which yields a 384-dimensional vector.

4 Training

4.1 Loss

4.2 Training procedure

5 Results

REFERENCES

[1] Sabrina Jaeger, Simone Fulle, and Samo Turk. Mol2vec: Unsupervised machine learning approach with chemical intuition. *Journal of Chemical Information and Modeling*, 2018. URL https://pubs.acs.org/doi/10.1021/acs.jcim.7b00616.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *arXiv (Cornell University)*, 2 2020. URL http://export.arxiv.org/pdf/2002.05709.

[3] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SIMCSE: Simple Contrastive Learning of Sentence Embeddings. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 1 2021. doi: 10.18653/v1/2021.emnlp-main.552. URL https://doi.org/10.18653/v1/2021.emnlp-main.552.

[4] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Píetro Lió, and Yoshua Bengio. Graph attention networks. *arXiv (Cornell University)*, 2 2018. doi: 10.17863/cam.48429. URL https://arxiv.org/pdf/1710.10903.pdf.

[5] Shaked Brody, Uri Alon, and Eran Yahav. How Attentive are Graph Attention Networks? *arXiv (Cornell University)*, 5 2021. doi: 10.48550/arxiv.2105.14491. URL https://arxiv.org/abs/2105.14491.

[6] Zhitao Ying, Jiaxuan You, Christopher J. Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *neural information processing systems*, 31: 4805–4815, 12 2018. URL https://arxiv.org/pdf/1710.10903.pdf.

[7] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MINILM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. *arXiv (Cornell University)*, 2 2020. URL https://arxiv.org/pdf/2002.10957.pdf.