ALTEGRAD Data Challenge Report
Molecule Retrieval with Natural Language Queries

Borachhun YOU
borachhun.you@ens-paris-saclay.fr

Haocheng LIU
haocheng.liu@polytechnique.edu

Ly An CHHAY
ly-an.chhay@polytechnique.edu

February 2024

1 Introduction

This challenge concerns a task of molecule retrieval by using natural language queries. In other words, given a text query, we want to retrieve a molecule, which is represented in graphs, that corresponds to the query from a given set. This has raised a challenging problem of integrating these two very different modalities (text and graph). Therefore, the objective of this challenge is to build a deep learning model that is capable of achieving this specific task. The performance of the model is to be evaluated on a test set given in a Kaggle competition, in which our team is named **"CAC"**.

2 Dataset

For the training and evaluation of the model, two different types of dataset were given: graphs representing the molecules and text descriptions. The molecule dataset consisted of a total of 102981 graphs, whereas the text dataset was divided into 3 subsets of 26408, 3301, and 3301 samples for training, validating and testing respectively. Each text description in the training and validation sets was matched with the ID of the corresponding graph, while this information was not given in the test set.

We visualize different characteristics of textual description and molecule graph in the dataset as Fig 5 in Appendix. The description length distribution indicates that textual descriptions are typically concise, with a right-skewed distribution that suggests most descriptions are short, but there are outliers with much longer lengths. The node count and degree distributions reveal a prevalence of sparsely connected graphs, with most nodes having only one or two edges. Additionally, the molecule graphs generally have a small number of nodes, pointing to a dataset dominated by smaller graphs. The clustering coefficient distribution, displayed on a logarithmic scale, shows that nodes are less likely to form tightly knit clusters, with most graphs having lower clustering coefficients. Finally, the average shortest path length distribution suggests that the graphs are relatively compact, with most nodes being separated by only a few edges, although there is variability, and some graphs exhibit longer paths between nodes. Overall, the dataset is characterized by smaller, sparser, and more loosely connected graphs, with a few exceptions that present more complexity, which requires the model to have a robust structure to capture these features to give reasonable embedding.

3 Model architecture

In order to accomplish the text-to-graph retrieval task, two different encoders were required in order to produce the representation of each of the data type.

3.1 Graph encoder

To encode the graphs, we tested multiple architectures in an attempt to find the one that gives the best performance. Convolution-based GCN [6] and GraphSAGE [5] as well as attention-based GAT [9] were some models that were considered. Based on evaluations on the test set, GAT outperformed the other models and it was chosen as the architecture for the encoder.

After that, we proceeded to test different numbers of layers of the model. When evaluated on the test set, the performance appeared to be improved as the number of layers increased up until 10 layers, which was our final choice.

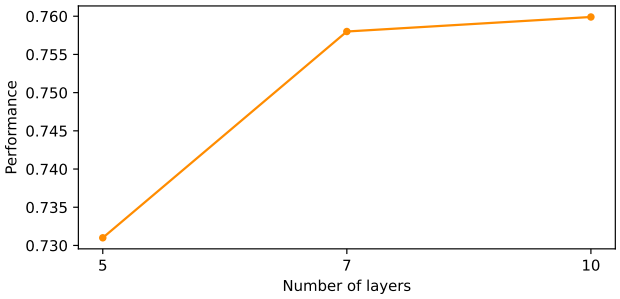


Figure 1: Performance on test set (with DistilBERT, 30 epochs)

In addition, residual connections were also included in the model. Specifically, the output of each layer was added with the input before being passed to the next layer.

3.2 Text encoder

In the process of encoding textual descriptions for each molecule, various pre-trained language models have been employed. These include DistilBERT, BERT Large, RoBERTa, and SciBERT.

- **DistilBERT** [8] is a lightweight version of BERT. It retains the essential features of BERT while using fewer parameters, making it computationally more efficient. DistilBERT serves as our baseline text encoder.
- **BERT Large** [3] denoting a more extensive version of BERT with heightened layer and parameter counts, aims to capture intricate relationships within textual data. Our investigation explores the potential benefits of employing a larger and more exhaustive model in capturing and embedding the nuances of our molecule descriptions.
- **RoBERTa** [7] represents a robust and optimized variant of BERT, exhibiting state-of-the-art performance across diverse natural language processing benchmarks and tasks. Our evaluation includes testing this model to assess its effectiveness in capturing the semantics embedded within our molecule text descriptions.
- **SciBERT** [2] a pre-trained language model based on BERT , trained on a large multi-domain corpus of scientific publications to improve performance on downstream scientific NLP tasks. Given the relevance of our challenge to science , specifically biochemistry, we consider SciBERT as a potential candidate to enhance performance in our study.

Employing various text encoders in conjunction with GAT as our graph encoder, and based on evaluations conducted on the test set, our findings indicate that SciBERT demonstrated superior performance compared to the other text encoder models. The baseline encoder, DistilBERT, secured the

second position with a marginally lower LRAP score. In contrast, despite their more intricate architectures, BERT Large and RoBERTa lagged behind in performance. Therefore, we will mainly focus on the performance results of DistilBERT and SciBERT in the experimental result section.

4 Training

The two encoders of the model are trained simultaneously through contrastive learning. This method involves learning similarities between the texts and the graphs in the representation space.

4.1 Loss

The baseline provided implements a form of contrastive loss using cross-entropy loss, but it's conceptually different from the traditional contrastive loss formula, which aims to ensure that graph embedding is similar to the corresponded textual description embedding than to any other embedding in the set.

Given a batch of encoded embeddings $\mathbf{e}_{txt}$ and $\mathbf{e}_{graph}$, where $\mathbf{e}_{txt}$ and $\mathbf{e}_{graph}$ are each matrices of size (N, D), with N being the number of embeddings in the batch and D being the dimensionality of each embedding, the goal is to make each embedding in $\mathbf{e}_{txt}$ as close as possible to its corresponding embedding in $\mathbf{e}_{graph}$ (and vice versa), and distant from all other embeddings. The similarity score between embeddings is calculated using the dot product, resulting in a similarity matrix $\mathbf{S}$ of size (N, N), where each element s_{ij} is given by:

$$s_{ij} = \mathbf{e}_{txt}^{i} \cdot {\mathbf{e}_{graph}^{j}}^{T} \tag{1}$$

where $\mathbf{e}_{txt}^{i}$ is the i-th row of $\mathbf{e}_{txt}$ (i.e., the i-th embedding in $\mathbf{e}_{txt}$), and $\mathbf{e}_{graph}^{j}$ is the j-th row of $\mathbf{e}_{graph}$. The transpose (T) is used to calculate the dot product.

The cross-entropy loss for each row of $\mathbf{S}$ with respect to the correct matching embedding is then calculated. Assuming that $\mathbf{e}_{txt}$ and $\mathbf{e}_{graph}$ are two same embeddings representing same physical molecules, hence the diagonal of $\mathbf{S}$ should ideally show the highest scores. The contrastive loss using cross-entropy can be expressed as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp(s_{ii})}{\sum_{j=1}^{N}\exp(s_{ij})}\right) - \frac{1}{N}\sum_{j=1}^{N}\log\left(\frac{\exp(s_{jj})}{\sum_{i=1}^{N}\exp(s_{ji})}\right)$$

This loss function forces the model to increase the similarity scores on the diagonal of $\mathbf{S}$ (matching embeddings) relative to off-diagonal elements (non-matching embeddings), effectively learning embeddings from both textual description and molecule graph.

4.2 Optimizer

The optimizer that was used to train the model was AdamW. For regularization, we tested a few different values for the weight decay parameter and 0.01 gave the best result. As for the learning rate, the exponential learning rate scheduler was utilized in order to obtain variable learning rates throughout the training epochs. The learning rate can be computed by:

$$l_n = l_0 \times e^{-kn}$$

where l_n is the learning rate at iteration n and k is the decay parameter which was set to 0.9. The initial learning rate was set to $l_0 = 2 \times 10^{-5}$.

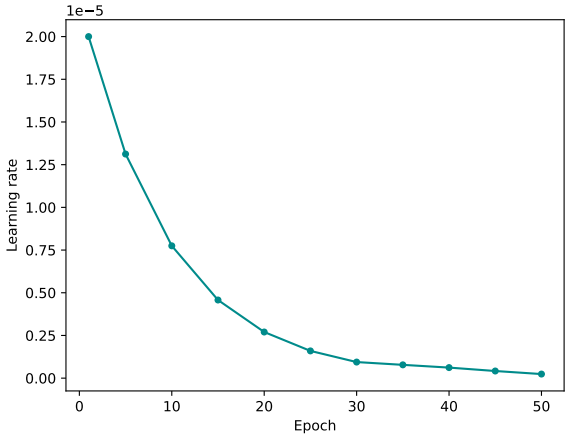


Figure 2: Learning rate changes (with DistilBERT and 10 layers of GAT)

Regarding the duration of the model training, we observed that the more epochs we trained the model for, the higher the performance was, which was true up until 50 epochs or even 60 epochs in some cases.

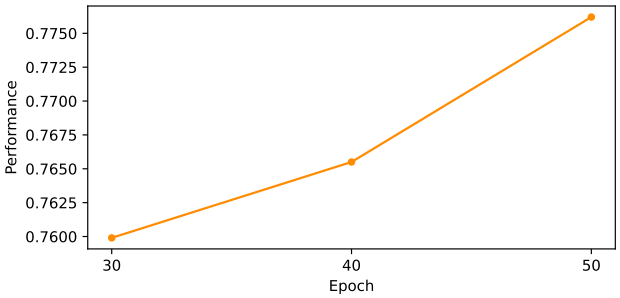


Figure 3: Performance on test set (with DistilBERT and 10 layers of GAT)

5 Experimental results

The table below depicts the performance results of the different models we have implemented.

Model	Validation loss	Test score (Leaderboard)
DistilBERT + GCN3 (Baseline)	0.3740	0.4897
DistilBERT + GAT5	0.3000	0.6087
DistilBERT + GAT5 + Residual	0.2700	0.6654
DistilBERT + GAT10 + Residual	0.2601	0.7762
SciBERT + GAT10 + Residual	0.1016	0.8000

Table 1: Comparison of Model Performance

The introduction of the attention-based architecture of Graph Attentional Network (GAT) as the graph encoder to our baseline model resulted in a notable enhancement in LRAP score, progressing

from 0.4897 to 0.6687. The incorporation of Residual connections further contributed to the improvement by addressing the vanishing gradient issue, resulting in a refined score of 0.6654. Subsequently, increasing the GAT layer count to 10 augmented the depth of the graph neural network, allowing the model for a more intricate capture of the complex relationships within the molecule's node and edge structures. This adjustment led to a substantial increasing score of 0.7762 in the test set. Finally, aligning with the scientific focus of our text data nature on Biochemistry, the substitution of DistilBERT with SciBERT demonstrated improved performance, culminating in a score of 0.8.

Furthermore, the validation loss of each model decreases gradually. The decreasing trend in validation loss serves as a corroborative indicator of the efficacy of the introduced modifications, reaffirming the models' improved capacity to accurately capture and represent the intricate patterns within the given data. This alignment between diminishing loss and augmented performance underscores the effectiveness of the implemented enhancements in refining the models' learning capabilities and overall predictive accuracy.

6 Further improvement

In our study, we considered designing a discriminator to enhance the learning of text and graph networks. Due to the modal differences between molecular graphs and textual descriptions, primarily in terms of data representation and semantic alignment, issues related to cross-modal integration and information inconsistency have arisen. The integration of a discriminator can simplify the alignment process by effectively distinguishing between the features of the two modalities. Thus, in this section, we will introduce this experiment and explore its implications for improving the coherence and efficiency of learning from both textual and graphical data sources.

We are actually generating both text and graph embedding, which could be imagined as a generative modeling task. Thus it might be possible to introduce a discriminator (like in the classic GAN-based model) to balance the cross-modal embedding. One of the most recent work [10] introduces a novel Adversarial Modality Alignment Network and gets excellent performance on Molecule information retrieval tasks, in which they use a Wasserstein metric guided GAN [4] as their adversarial network loss.

When we simply incorporated a discriminator (Also called Critic in WGANs [1]) to estimate the Wasserstein distance between encoded embeddings, and took it as another optimization objective, the model's training performance was improved. Fig 4 demonstrates that the incorporation of the adversarial structure has made the training more effective. After nearly 200 iterations, the training loss became lower than that of the larger language model SciBERT and remained below it. This confirms the discriminator's aid in the model's learning, accelerating the model's convergence. However, it is worth mentioning that the generalization performance on the validation set of the adversarial model (Leaderboard Score=0.76) was not as good as SciBERT's (Leaderboard Score=0.80). Due to computing resource limitations, more training (SciBERT-Adversarial) could not be completed, but we consider it an interesting attempt and believe it to be a promising method for the future.

7 Conclusion

Throughout this report, we have presented our approach of tackling the given problem of molecule retrieval with natural language queries. We started by studying the provided datasets, before presenting different graph and text models that were tested and eventually our final choices. Then, we discussed the training procedure, mainly on the loss and the optimizer, before giving the performance results.

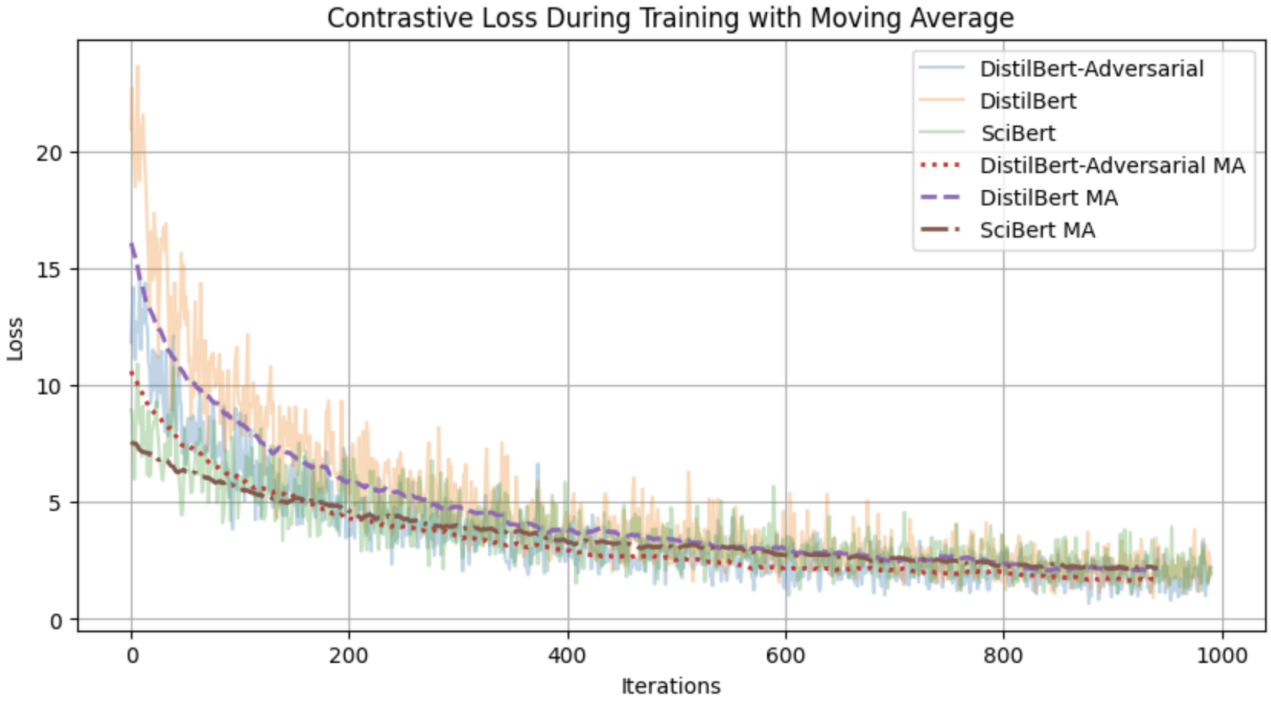
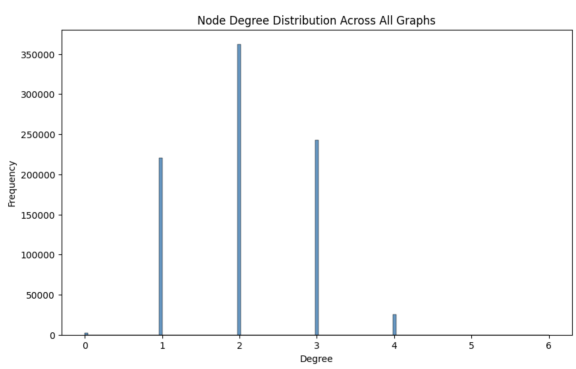


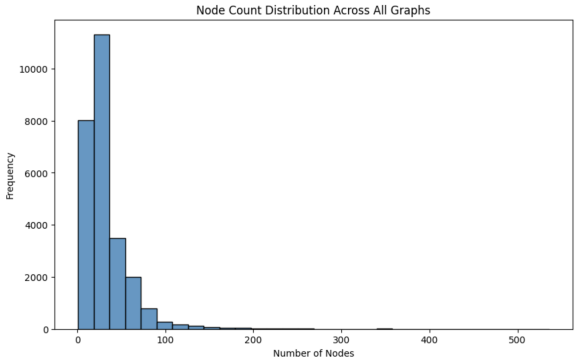
Figure 4: Contrastive Loss During Training with Moving Average

The ultimate model in our study, achieved by integrating SciBERT as the text encoder and utilizing a Graph Attentional Network (GAT) with 10 layers and residual connections, exhibited a remarkable score of 0.8 on the validation set. To enhance the model further, the incorporation of a discriminator into the architecture (SciBERT-adversarial) is anticipated to yield more promising results.

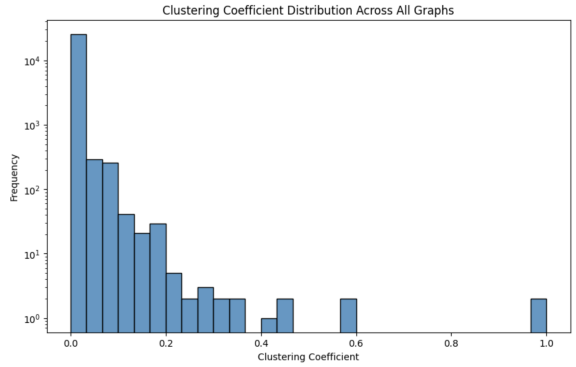
8 Appendix



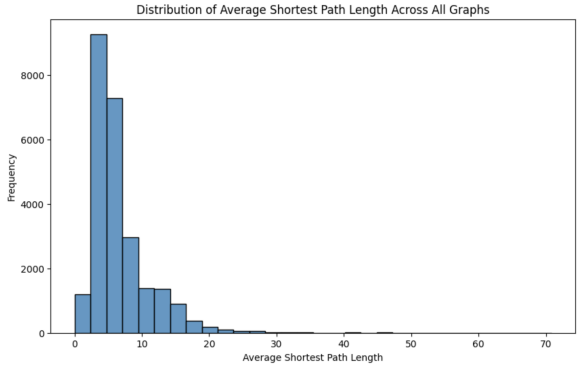
(a) Degree Distribution



(b) Degree Distribution



(c) Clustering Coefficient Distribution



(d) Average Shortest Path Length Distribution

Figure 5: Molecule Graph Characteristics

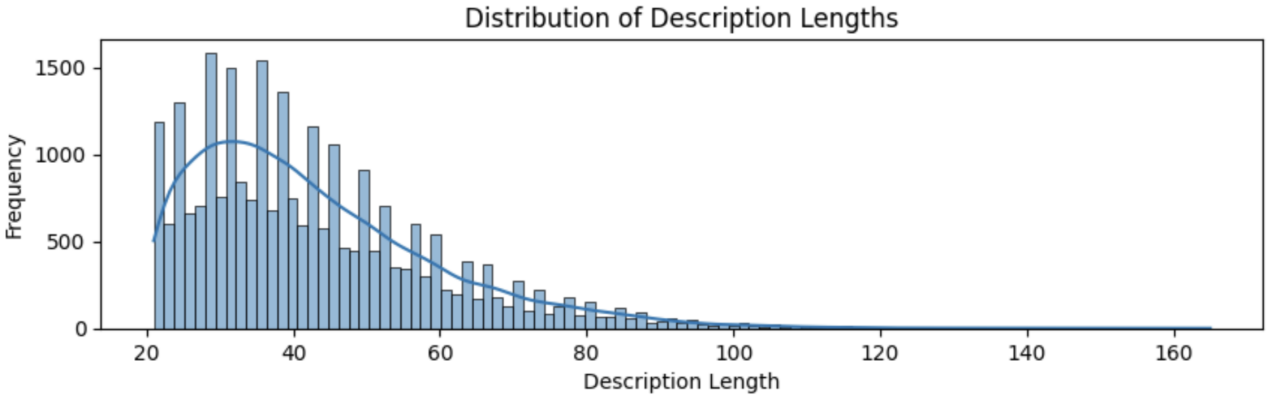


Figure 6: Textual Description Length Distribution

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In *EMNLP*. Association for Computational Linguistics, 2019.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017.

[5] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018.

[6] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[8] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[9] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[10] Wenyu Zhao, Dong Zhou, Buqing Cao, Kai Zhang, and Jinjun Chen. Adversarial modality alignment network for cross-modal molecule retrieval. *IEEE Transactions on Artificial Intelligence*, 5(1):278–289, 2024.