ROVERATO Chiara
chiara.roverato@student-cs.fr

BEN ZENOU Gabriel
gabriel.benzenou@student-cs.fr

Molecule Retrieval with Natural Language Queries

MVA - ALTEGRAD

02/04/24

1 Introduction

Generation of molecules based on desired proprieties, exploration of databases with textual description or any other task on molecules' graphs and text are becoming possible thanks to the recent advances in Artificial Intelligence (AI) for Natural Language Processing (NLP) and Graphs processing. For the ALTeGRaD course from the MVA master in ENS Paris-Saclay, we took part in the Kaggle competition ALTeGRaD-2023 Data Challenge we implemented a multimodal model able to capture similarities between molecules' graphs and their textual description.
To this end, we had at our disposal a bunch of various data:

- a training and a validation dataset composed of textual descriptions paired with their tokenized molecule graphs;
- a test dataset composed of a list of textual descriptions and a list of tokenized molecule graphs;
- a dictionary whose keys are the molecule graphs tokens and whose values are corresponding embeddings, computed with the mol2vec model.

In addition to this data, we started with a baseline model made with the architecture showed in 1. The text tokenizer and encoder are issued from the DistilBERT model, a lighter version of BERT, easy to train locally [6]. The graph encoder is a GCN with three layers.

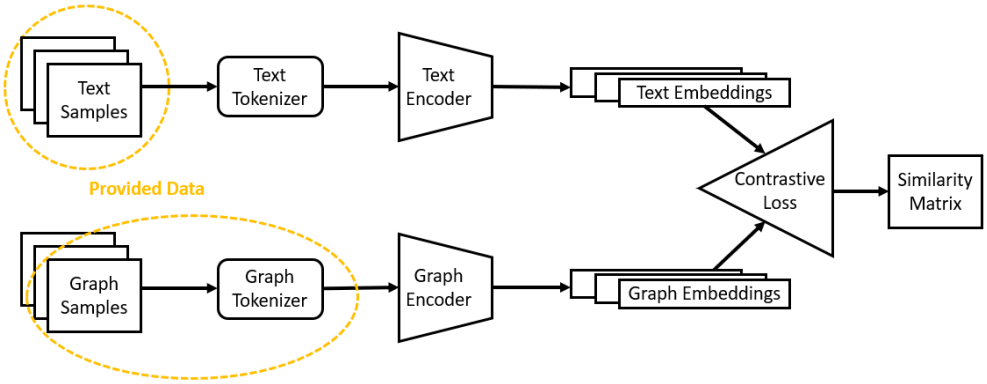

Figure 1: Baseline model's architecture

To get improved predictions, we decided to play on four main elements of the base pipeline:

- the general architecture;
- the text model;
- the hyper parameters of the overall model;
- the loss.

2 Our approach

2.1 Tokenizer and Encoder

The first path we explored was improving the text tokenizer and encoder pre-trained models. Even though DistilBERT was vastly used in the litterature, [4] uses another model, SciBERT. It is a model that was trained on scientific data, which matches perfectly to our job and, according to [2], SciBERT outperforms DistilBERT on scientific data.

On a second thought, as we plan on making the final predictions with an ensemble model, we thought we might also gain in accuracy by trying a non-scientific model to capture the usual semantics behind the textual captions. For this purpose, we looked for the best text encoder according to litterature's benchmarks and we found RoBERTa was quite promising [3].
However, SciBERT and RoBERTa's downside is that they are way heavier than DistilBERT and thus cannot easily be run locally.

2.2 RUCHE's hardware

In order to train our model using SciBERT and RoBERTa pre-trained models, it was necessary to get access to strong computational devices. Being CentraleSupélec students, we were able to connect to the school supercomputer RUCHE [1].
Based on the Slurm framework, RUCHE gave us access to Nvidia Tesla V100S GPUs.

2.3 Learning Rate and Dropout Search

In order to further optimise our model, we decided to try to find the best parameters for the task. We first trained the model on 5 epochs for different learning rates, without any dropout 2. Initially, the learning rate was set at 2×10^{-5}. We discovered that the best validation loss was obtained for a learning rate of 5×10^{-6}, resulting in a division by 6 of the validation loss.

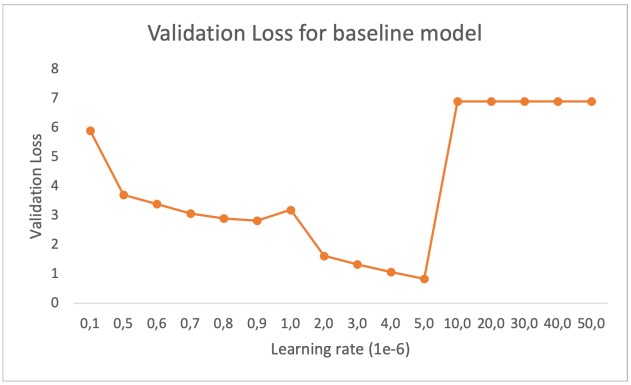

Figure 2: Learning rate influence on the validation loss

To avoid over-fitting and improve our model's performance we added a dropout in the graph and text encoder. We proceeded as before to find the best parameter: for a fixed learning rate of 1×10^{-6}, we tuned the dropout rate. The optimal dropout rate was a dropout of 0.1 3.

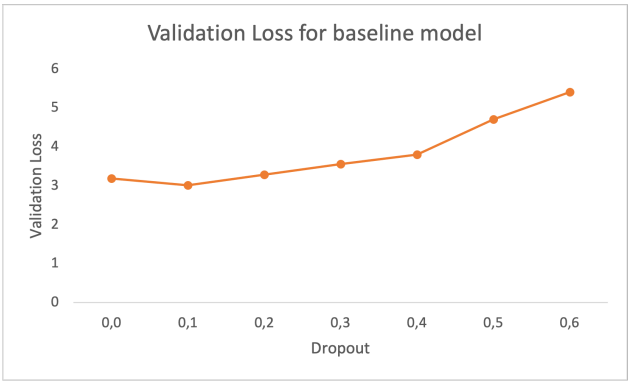

Figure 3: Dropout influence for a learning rate of 1×10^{-6}

With these changes, training our model on 20 epochs, for a learning rate of 5×10^{-6} and for a dropout rate of 0.1, we improved our previous LRAP score from 0.48 to 0.51. Then, training it on the training and validation dataset, for 28 epochs, we achieved a LRAP score of 0.55.

2.4 Ensemble Model

As a final ingredient to our recipe, we built the similarity score matrix on the test dataset from several different models:

- RoBERTa avec $lr = 5e-6$ et $dropout = 0$
- RoBERTa avec $lr = 5e-6$ et $dropout = 0.1$
- RoBERTa avec $lr = 5e-6$ et $dropout = 0.2$
- SciBERT avec $lr = 5e-6$ et $dropout = 0$
- SciBERT avec $lr = 5e-6$ et $dropout = 0.1$
- SciBERT avec $lr = 5e-6$ et $dropout = 0.2$

We expect such a variety of hyperparameters and text tokenizers and encoders to capture different semantics behind the text and graph samples.
To build the ensemble similarity matrix, we tried two different averaging method:

- Firstly, we computed the classic average of all the individual similarity matrices.
- Secondly, we computed the average of the softmax of all the individual similarity matrices. We chose to apply a softmax before taking the average in the event that we can trust each model for giving a high similarity score to a pair only if this pair is valid.

2.5 Results

The results are described in 4. We see that our implementation of the ensemble model is quite disappointing and doesn't bring high value to our model.

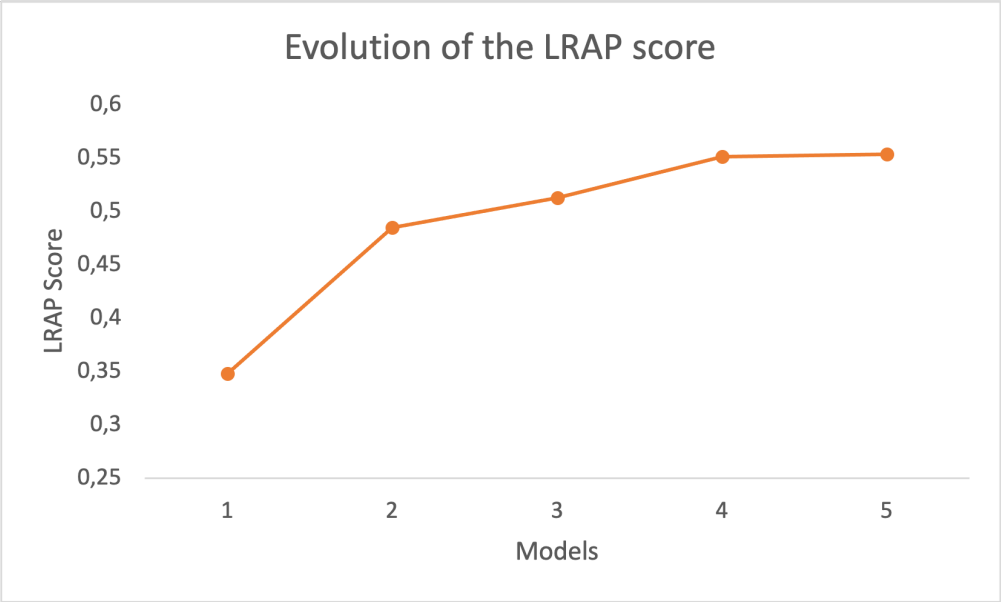

Figure 4: Evolution of the LRAP score for our different models

3 Other Approaches

3.1 Contrastive Loss Approach

Reading the paper the paper [5], we decided to implement on other typè of loss for the contrastive loss: the EBM-NCE loss, defined as follows:

$$L_{EBM-NCE} = -\frac{1}{2}\left(\mathbb{E}_{x_c,x_t}\left[\log\sigma(E(x_c,x_t))\right] + \mathbb{E}_{x_c,x'_t}\left[\log(1-\sigma(E(x_c,x'_t)))\right]\right.$$
$$\left.+\mathbb{E}_{x_c,x_t}\left[\log\sigma(E(x_c,x_t))\right] + \mathbb{E}_{x'_t,x_c}\left[\log(1-\sigma(E(x'_t,x_c)))\right]\right)$$

where σ is the sigmoid activation function, x_c and x_t form the structure-text pair for each molecule, and $x_{c'}$ and $x_{t'}$ are the negative samples randomly sampled from the noise distribution, which we use the empirical data distribution. $E(\cdot)$ is the dot product on the jointly learned space. This loss uses the BCE With Logits Loss wich is a version more numerically stable than using a plain Sigmoid followed by a BCELoss by combining the operations into one layer.

Unfortunately, this solution did not improve our result compared to the original contrastive loss and we decided not to further explore this option.

3.2 Cross-Modal Approach

3.2.1 Architecture

As presented in introduction, we read the [4] paper and decided to implement its cross modal approach.
The baseline of our model was conceived of two "towers", one for each modality: on the first tower, we embed text descriptions, and on the second tower, we embed molecule graphs. The idea is to "build a bridge between these two towers", by adding a transformer decoder layer with targets being the graph tokens and sources being the text tokens (see fig5). This way, text descriptions are embedded conditionally to their corresponding graph.

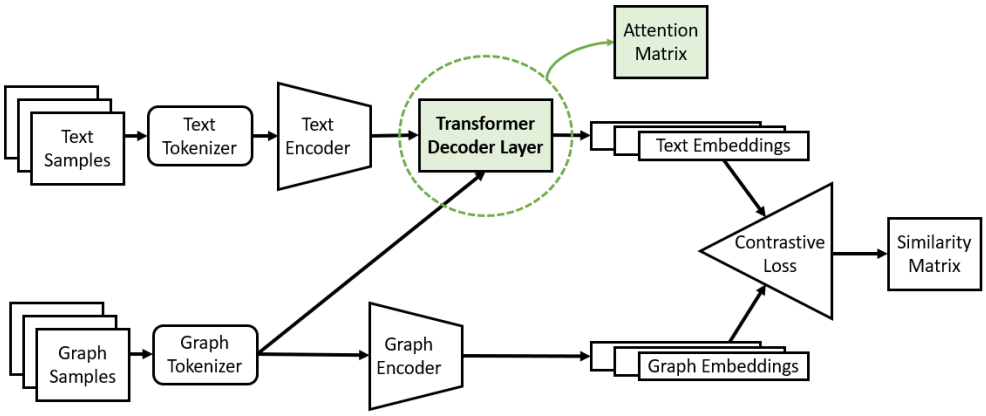

Figure 5: Cross-modal model's architecture

3.2.2 Training

With such an architecture, if a similar training as before is performed, the optimal way for the model to train is without taking into account the text descriptions of each pair. Indeed, to minimize the contrastive loss, the added transformer decoder layer will just train to hide the text descriptions and keep only the graph input. This way, the graph data will leak into the text tower and, eventually, the model will output on the one side the graph embedding from the graph tower, and on the other side the same graph embedding from the text tower. Which is not the behaviour we expect.
To counter this effect, we implemented a negative-paired training process: in the training loop, when getting the batch from the dataloader, we randomly select half of the batch to modify its graph inputs into other graph inputs. We expect this training process to force the model into learning text representations rather than being blind to it.

3.2.3 Association Rules and Prediction

Once the cross-modal model is trained, we can use it. However, we can't simply run it on the test dataset as we would need to know in advance which text sample correspond to which graph sample, as the text embeddings are generated conditionally to the graphs they correspond to.
Instead, we base our prediction technique on the attention weights of the transformer decoder layer. We perform inference with the cross-modal model on the training and validation datasets and we save for each samples the output attention weights matrix from the transformer decoder layer. Such matrix is of size $(number\_of\_text\_tokens, number\_of\_graph\_tokens)$ and we interpret it as a "probability" (it's not exactly a probability as the weights do not sum to one) for a text token and a graph token to match one with the other. Once saved, we loop over all graph and text tokens and create an "Association Rules" dictionary which contains

for each graph token and text token pairs the sum of all the corresponding attention weights. This dictionary now allows us, for a given text sample and graph sample pair, to give a similarity score that we call confidence. We thus compute the confidence matrix for all graph samples with all text tokens, and we can submit it on kaggle.

3.2.4 Limits

Although the whole "cross-modal idea" presented in the above is issued from [4], we implemented all the code by ourselves.
Eventually, we were able to train the model with the negative-paired training technique above mentioned and we got a good looking loss for 18 epochs (we weren't able to compute the rest of the epochs due to time limit on RUCHE) (see Fig6).

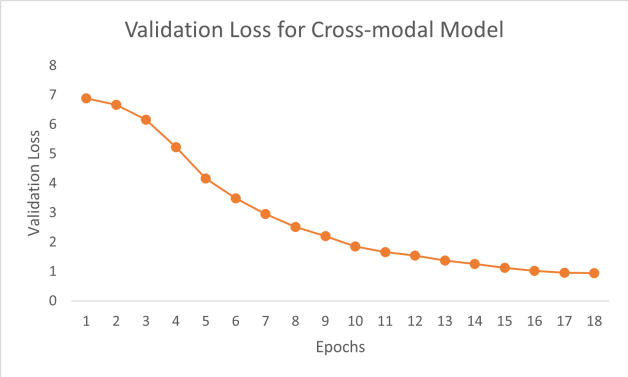

Figure 6: Evolution of the validation loss during the Cross-modal model training

However, when observing the attention weights for computing the association rules, we noticed that for a given graph token, all attention weights had a constant value, whatever the text token. Which would be easily explained if no negative-paired training process had been put in place. But we did implement a working negative-paired training process. We weren't able to understand what was the reason for such a behavior and we thus couldn't use it as prediction model.

4 Conclusion

Our work aimed at retrieving molecules using natural language queries, combining NLP and machine learning techniques. This project involved understanding and optimising each element at the model, trying to improve the tokenizer and encoder with SciBERT and RoBERTa, optimising the model's parameters like the learning rate and the dropout rate, improving the contrastive loss, and searching for additional approaches to better pair the molecules to their descriptions.

Although we managed to improve the global LRAP score of the model, additional steps would be to overcome the cross-modal's difficulties and to try new model architectures , with different layers or text encoders. We did not investigate to find better graph encoder but this may be a valuable path to explore.

This project allowed us to deeply dive into a real deep-learning problem, to face computer limitations, and to discover via the literature and our experience the vast field of multi-modal deep learning.

References

[1] Ruche overview - mesocentre documentation - https://mesocentre.pages.centralesupelec.fr/user$_{d}oc/ruche/01_{c}luster_{o}ver$

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text.

[3] Arjun Bhatt, Ruth Roberts, Xi Chen, Ting Li, Skylar Connor, Qais Hatim, Mike Mikailov, Weida Tong, and Zhichao Liu. DICE: A drug indication classification and encyclopedia for AI-based indication extraction. 4.

[4] Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2mol: Cross-modal molecule retrieval with natural language queries. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607. Association for Computational Linguistics.

[5] Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Animashree Anandkumar. Multi-modal molecule structure–text model for text-based retrieval and editing. 5(12):1447–1457. Number: 12 Publisher: Nature Publishing Group.

[6] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.