JOHANNES KEPLER
UNIVERSITY LINZ

Chemformer: A Pre-Trained Transformer for Computational Chemistry

This is a report on the project of implementing the Chemformer repository published by: Ross Irwin1 , Spyridon Dimitriadis1,2, Jiazhen He1 , and Esben Jannik Bjerrum1

Author
Mostafa Mohamed Abdelrazek

Submission
Institute for Machine Learning

Supervisor
Dr. Philipp Seidl, MSc

June 1, 2024

Practical work report
For the academic degree of
Master's of Artificial Intelligence

**JOHANNES KEPLER
UNIVERSITY LINZ**
Altenberger Straße 69 4040
Linz, Austria

Abstract

This report documents the setup and application of the Chemformer repository for molecular sequence modelling using Transformer-based architectures. The work involved adapting and debugging an older codebase running on a modern Ubuntu system with single-GPU support to take advantage of multiworkers option. The main target was to fine-tune a pretrained Chemformer model and applying molecular predictions to evaluate the performance. But since the last check point was already really good no new checkpoint was produced during fine-tuning, so that last checkpoint was used for prediction and evaluation tasks. Outputs were generated as multiple candidate SMILES for each target with the likelihood as a confidence scale for prediction, revealing areas for improvement, such as canonical SMILES matching and molecule equivalence scoring. This work provides a solid foundation for further exploration of Chemformer's capabilities in a limited-resource environment.

Introduction

Recently there was many researches applying neural network models to cheminformatics tasks. Sequence-to-sequence models, such as the Transformer [1] and models based on the Recurrent Neural Network (RNN) architecture [2, 3], are well suited to tasks such as direct reaction prediction, retrosynthesis prediction and molecular optimisation. Applying molecules encoded using Simplified Molecular Line Entry System (SMILES) [4] to the Transformer model has produced state-of-the-art results on benchmark datasets for these tasks [5,7]. Transformers have also been successfully applied to discriminative tasks such as biological activity prediction (virtual screening) [8] and molecular property prediction (QSAR modelling) [8,14]. Training Transformer models on SMILES strings, however, can be computationally expensive; a recently proposed model for direct synthesis prediction requires two days of training [15]. Additionally, separate models must be built, trained and tuned for each task, increasing the amount of effort required by research teams

Self-supervised learning using the Transformer has revolutionised Natural Language Processing (NLP) in recent years; large language models such as BERT [16], BART [17], GPT [18, 19], UniLM [20] and T5 [21] have provided significant improvements on key benchmark NLP tasks. Pre-training these models – training on a large unlabelled dataset of text before fine-tuning on the dataset of interest – has been shown to improve results on downstream tasks, and that is what used in this project.

Inspired by advances in Natural Language Processing, this project aims to address resource limitations in molecular modeling by leveraging transfer learning through

Transformer-based architectures. SMILES is treated as a "chemical language," [4] enabling the application of language models to a unified molecular representation. The core objective is to evaluate how self-supervised pre-training on large, unlabeled molecular datasets can accelerate convergence and improve performance in low-resource settings. We focus on fine-tuning a pretrained Chemformer model and conducting predictions on a constrained dataset due to hardware limitations. Through this, we demonstrate the practical benefits of transfer learning for molecular prediction tasks and highlight its potential in enhancing the efficiency of cheminformatics workflows, even in environments with limited computational resources.

Repository Setup

The Chemformer repository was deployed on a local Ubuntu system. However, due to the outdated nature of the repository and the absence of a comprehensive `requirements.txt` file, the setup process required manual intervention. Many dependencies were either deprecated or incompatible with current library versions. This included:

- resolving version conflicts, and adapting script syntax to be compatible with modern Python and PyTorch environments, PyTorch Lightning, and Hydra
- factor some utility functions and scripts to ensure compatibility with the hardware setup, specially CUDA libraies to inable GPU usage

These efforts were crucial for the full functionality and ensuring the codebase could be successfully used for model fine-tuning and prediction tasks in a constrained computational environment.

Selection of Tasks

Chemformer provides scripts for the following key tasks:

- `pretrain.py`
- `fine_tune.py`
- `predict.py`
- `inference_score.py`
- `build_tokenizer.py`

The main Chemformer research required too much effort to be done and including many computations and high performance hardware capabilities, so better reinvent the wheel from the beginning it was more practical to start from the best models trained and finetune it by `fine_tune.py` hopping to improve the last checkpoint then applying `predict.py` as a way of comparison. Also those two tasks required the lowest computational time comparing with the other tasks.

Model, Data and Processing Constraints

Based on the tasks selection the best suted model for fine tuning was the "uspto_50 " as it is used for forward reaction prediction, taking the "reactants" in SMILES form as input and predict the "products" in SMILES too. Selecting uspto_50 for prediction recuired choosing sequence-to-sequence data because the task involves translating a chemical input (typically a set of reactants or reagents in SMILES format) into a desired output (usually a product or a set of reactants), just like in machine translation. Although i have chosen the lowest tasks in computation need, it required too much computation considering the available hardware so it was crucial to do preprocessing before fine-tuning and prediction, as:

- Subsets of data were created and processed (e.g., 1000 samples)
- `limit_input.py` script was written to extract manageable slices
- The Ubuntu platform was leveraged to take advantage of the multiworkers option so the CPU was used for data preparation before passing it to the GPU

The trick of the multiworkers made the computation more efficient but it would have not make any difference without slicing a manageable subset of the data.

Fine-Tuning

The idea was to fine-tune the last checkpoint recorded for the uspto_50 trying to improve it by adjusting the hyperparameters, but the last checkpoint recorded was already good, unfortunately there was no improvement. And since ModelCheckpoint in PyTorch Lightning only saves if `save_top_k > 0` and `monitor` metric improves, so i couldn't produce a new checkpoint. But it is understandable that there was no improvement because it was impossible to use the whole dataset for fine-tuning as it requires an advanced hardware, and the sliced data subset was not enough to make a difference.

Prediction and Evaluation

Since there were no checkpoints generated from the fine-tuning, an existing pretrained checkpoint was used. Predictions were performed using the `predict.py` script from the Chemformer repository.

The model took target SMILES strings as input and generated multiple ranked candidate SMILES as output. Each candidate was accompanied by:

- log-likelihood score, indicating the model's confidence in that prediction
- The output data was structured in tabular form, with each row containing the target SMILES
- its top sampled predictions, and the corresponding log-likelihood values

Initial performance evaluation focused on Top-1 and Top-10 accuracy metrics. However, the match rates were relatively low, which could be attributed to the syntactic variability of SMILES strings as different SMILES can represent the same molecule, also the performance evaluation should be applied after using the full dataset, so we can not tak that performance evaluation into account as we only used a small subset of the data.

Discusion

As a result, if we used the whole dataset, string-based comparisons may underestimate model performance, and future evaluations should consider molecule-level comparisons using cheminformatics tools such as RDKit to assess structural equivalence, which would provide a more accurate assessment of predictive quality.

Lack of proper documentation and dependency management consumes lots of time as future users should be aware of the need for manual environment setup. Also it is required to have an advanced Hardware or even to have an online source of GPU to be able to implement the tasks properly and get a meaningfull results and that what is planned in the future work.

Conclusion

Despite the initial challenges related to environment setup and limited computational resources, this project successfully applied fine-tuning and prediction tasks using the Chemformer model. The work constructed a robust technical foundation for further exploration of Transformer-based approaches in cheminformatics. With the infrastructure now functional, we have a good base to explore more capabilities of this model or even extend it to more tasks such as molecular property prediction, masked SMILES reconstruction, and more accurate evaluation using canonical molecular fingerprints or structure-based metrics. This provides a foundation for more extensive research and the potential for contributions to low-resource molecular modelling applications.

Refences

[1] Vaswani, A. et al. Attention is all you need. In NIPS (2017).

[2] Hochreiter, S. & Schmidhuber, J. Long short-term memory. Neural computation 9, 1735–1780 (1997).

[3] Cho, K. et al. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734 (2014).

[4] Weininger, D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. Journal of chemical information and computer sciences 28, 31–36 (1988).

[5] Tetko, I. V., Karpov, P., Van Deursen, R. & Godin, G. State-of-the-art augmented nlp transformer models for direct and single-step retrosynthesis. Nature communications 11, 1–11 (2020).

[7] He, J. et al. Transformer neural network for structure constrained molecular optimization. ChemRxiv (2021).

[8] Fabian, B. et al. Molecular representation learning with language models and domain-relevant auxiliary tasks. arXiv preprint arXiv:2011.13230 (2020).

[14] Ross, J. et al. Do large scale molecular language representations capture important structural information? arXiv preprint arXiv:2106.09553 (2021).

[15] Schwaller, P. et al. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. ACS central science 5, 1572–1583 (2019).

[16] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).

[17] Lewis, M. et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 7871–7880 (2020).

[18] Radford, A., Narasimhan, K., Salimans, T. & Sutskever, I. Improving language understanding by generative pre-training (2018).

[19] Radford, A. et al. Language models are unsupervised multitask learners (2019).

[20] Dong, L. et al. Unified language model pre-training for natural language understanding and generation. arXiv preprint arXiv:1905.03197 (2019).

[21] Raffel, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research 21, 1–67 (2020).