Cost-aware Finetuning on Pretrained LM to Efficiently Perform Retrosynthesis of Chemical Reactions

Hongjia Huang
New York University
hh3043@nyu.edu

Zhaodong Liu
New York University
zl4789@nyu.edu

Haoqi (Kevin) Zhang
New York University
hz3223@nyu.edu

Abstract

SMILES (Simplified Molecular Input Line Entry System) is a sequential representation of molecules. With the help of language models, the task of chemical reaction prediction can be better approached. Our aim is to use less data from actual lab experiments and computational resources in fine-tuning to achieve solid performance on the chemical reaction prediction task by fully utilizing the benefits of small language models. Our project proposes an effective dataset construction methods along with investigation on a series of fine-tuning methods such as fixing parts of the model, adding MLPs, and LoRA.

1 Introduction

With the help of sequential representation such as SMILES, molecules could be understood in a way that is similar to text. Aided by the power of language models, we could use NLP techniques to learn about chemical reactions. In our work, we focus on the particular task of chemical reaction prediction problems: retrosynthesis (RS), which is to predict the reactants on the basis of the products of chemical reaction.

While currently, there exist some large language models for this task, they require too much cost in training and inferencing (in terms of time and computation) and are hard to optimize. Therefore, we tried to fully utilize a small scale language model instead. We divide our work into 3 parts: 1. proposing a data reconstruction methods that helps in creating dataset with shorter time and lower cost. 2. investigate on how our proposed data collection methods may effect the performance of full parameter fine-tuned model. 3. Try various methods in fine-tuning task to improve the model performance.

In summary, our work shows that leveraging LLM-derived data is a viable way to mitigate data scarcity, and that combining MLP or LoRA enhancements with partial parameter updates provides a cost-effective, high-performing alternative to traditional, fully fine-tuned models for retrosynthesis prediction.

2 Related Work

Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction[1]. The model treats reaction prediction as a machine translation problem between SMILES of reactants, reagents, and the products. It introduced a multi-head attention Molecular Transformer model that outperforms previous algorithms at that time. Molecular Transformer makes predictions by inferring the correlations between the presence and absence of chemical motifs in the reactant, reagent, and product present in the data set. This shows us that SMILES string as a sequential representation of molecules, can be used as input to Transformer based model and acquires good result in chemical reaction prediction task.

LlaSMol models on the other hand are a series of state-of-the-art models that is based on large language models and achieve top performance on a series of tasks.[2] With the help of large language model. This shows that LLM has the capacity in modeling more complex and variant chemical reactions. These papers are consistent with our goal of fully utilizing the capacity of smaller language model with the help of large language model.

Chemformer, a model based on the BART architecture, has demonstrated its suitability for sequence-to-sequence tasks, particularly in the domain of chemical reaction prediction, as highlighted in [3]. The paper shows the robustness of Chemformer as well as the model's competitive performance against existing ones. By leveraging the strengths of Transformer-based architectures, Chemformer represents a pivotal step towards more accurate and scalable chemical reaction modeling. Furthermore, the small scale LM offers a model pretrained on SMILES string reconstruction task which we would further fine-tuned on downstream retrosynthesis task.

3 Approach

We first propose a data collection method and then use a BART-based transformer model[3] which is pretrained on SMILES string reconstruction task to implement various fine-tuning methods.

In the data collection stage, we use LLM to predict the products of some expensive reactants or reactions that may take a long time. Then we reconstruct the dataset based on LLM prediction and the original dataset USPTO-50k[4].

In the fine-tuning stage, we use full parameter fine-tuning on the model as a baseline and test out various methods such as fixing middle or end layers of the model, applying MLPs and run Lora fine-tuning.

4 Experiments

4.1 Data

The dataset we are using is the USPTO-50K dataset, which is subset and preprocessed version of Chemical reactions from US patents (1976-Sep2016) by Daniel Lowe[4]. It includes 50K randomly selected reactions and is currently the main dataset used for chemistry machine learning research.

During our research process, we found that the current datasets are very limited. Most of them are variants of the original USPTO datasets. This is because actual lab experiments can be both slow and costly. So it would be quite difficult to construct or augment a dataset based on real experiments. To address this, we came up with our own dataset construction method.

Our construction method is: Given a partially constructed dataset, in this case the original USPTO-50K dataset[4], we use SOTA models (in our case LlaSMol[2]) to generate possible output data. Constructing a dataset is way faster than running the actual lab experiments, thus can potentially save resources and time.

Using this data construction method, based on USPTO-50K dataset, we also constructed our own dataset, called USPTO-50K_γ, where γ is the rate which chemical reaction output labels were replaced by LLM output, defined as

$$\gamma = \frac{\text{generated examples by LLM}}{\text{Total examples}}$$

We would then run our experiments on both the original USPTO-50K dataset in finetuning and the constructed USPTO-50K_γ dataset to investigate how LLM generated outputs effect the performance of LM.

4.2 Evaluation method

The main evaluation metrics we used are: token_accuracy and molecular_accuracy.

Token accuracy is defined as

$$\text{token\_accuracy} = \frac{\text{correct output tokens}}{\text{all output tokens}}$$

Molecular accuracy is defined as

$$\text{molecular\_accuracy} = \frac{\text{correct reaction predictions}}{\text{all predictions}}$$

For molecular_accuracy, we also included the accuracy of top K cases, we chose to report top 1, top 3, top 5.

4.3 Experimental details

We conducted experiments using the fine-tuning pipeline for the BART-based model configured for the retrosynthesis task. The model architecture comprises a 6-layer Transformer-based encoder and decoder, each with multi-head attention, feedforward layers, and LayerNorm components. The embeddings are of size 512, with an inner feedforward layer dimension of 2048. The training configuration included a learning rate of 0.001, a cyclic learning rate schedule, 100 epochs, and a batch size of 64 in the task of different gamma ratio, 128 in the task of trying methods such as LoRA and adding MLP layers. For different experiments, different treatments were applied, for example the encoder and decoder weights were frozen, and MLP layer was inserted between encoder and decoder during training. The experiments were conducted with dropout set to 0.1, using a pre-trained checkpoint. Our experiments all ran on NYU HPC, typically costs about 4 hours for the fine-tuning task with a single Nvidia RTX8000 GPU.

4.4 Results

We report the results in the following tables.

Table 1 is the the results of BART-based model run on constructed dataset USPTO-50K_γ with different gamma rates.

We observe that on the constructed dataset, as γ increases, the molecular level accuracy increases with lower token accuracy than those models trained on original dataset. This is possibly because the LLM is trained on much larger dataset and its prediction is based on all the knowledge, which is is much more molecule-predictable and also easier to be learned by the model in fine-tuning compared to the original dataset. This result reveals that the quality of the constructed dataset is solid and better than the original dataset, and can be used for training purposes, saving huge time and resources.

Table 2 is the results of BART-based model with different treatments applied, ran on the original USPTO-50K dataset.

Table 2 indicates that while token-level performance remains relatively stable across methods, achieving highest molecular accuracy requires full parameter fine-tuning. Freezing decoder weights is better than freezing encoder ones, and adding MLP layers after decoder further helps the model in achieving close performance in molecular accuracy. The model that have the closest performance to full parameter fine-tuning is Fixing decoder and adding MLPs after it.

Table 3 is the results of when the LoRA method is applied, with different treatments applied to different groups, ran on the original USPTO-50K dataset.

LoRA stands for "Low-Rank Adaptation". It's a parameter-efficient fine-tuning technique primarily used in transfer learning and large language models. It allows for updating a small number of parameters during fine-tuning by adding low-rank matrix updates to the original pretrained model's weights. This approach significantly reduces the computational and memory requirements. For our model, we only apply LoRA layer on the output projection layers of the Multi-head Attention block.

Table 3 indicated that when only using LoRA fine-tuning, the performance of the model is similar to the best one we obtained in Table 2. Furthermore, we observe that only adding MLPs after decoder still gives better performance as in Table 2. However, Fixing encoder gives better result than fixing decoder. Now, adding LoRA to our consideration, the best model we achieved in this case is: Fixing encoder, apply LoRA on decoder, and add MLPs after the decoder.

We also notice that the prediction accuracy in table 1 when gamma is 0 is higher than those of table 2, which is a result of different batch size used in the fine-tuning process. Due to hardware limitation (out of memory), we have to run the experiment of USPTO-50K_γ dataset on batch size of 64 instead

of 128 as we designed initially. As batch size can significantly influence training performance, the results are not directly comparable across table 1 and others.

In general, We find that using LLM-generated data can yield performance better than the original dataset. Moreover, introducing MLP layers or adopting LoRA-based fine-tuning strategies significantly improves results, nearly reaching full-parameter fine-tuning benchmarks.

Table 1: Model Performance Across Different Gamma Rates

Gamma	Test Mol Top 1 Acc	Test Mol Top3 Acc	Test Mol Top5 Acc	Test Token Acc
1.0	0.7418	0.8234	0.8449	0.9755
0.9	0.7272	0.8041	0.8281	0.9747
0.8	0.6973	0.7922	0.8115	0.9740
0.7	0.6832	0.7731	0.8045	0.9743
0.6	0.6744	0.7658	0.7893	0.9764
0.5	0.6553	0.7512	0.7707	0.9768
0.4	0.6260	0.7416	0.7611	0.9776
0.3	0.6354	0.7227	0.7535	0.9787
0.2	0.5930	0.7057	0.7455	0.9773
0.1	0.5967	0.7088	0.7357	0.9779
0.0	0.5879	0.6928	0.7359	0.9788

Table 2: Evaluation Metrics for Different Treatments without LoRA

Treatment	Test Token Accuracy	Top 1 Accuracy	Top 3 Accuracy	Top 5 Accuracy
Full Parameter	0.9783	0.5634	0.6840	0.7220
Fix Encoder	0.9571	0.2556	0.3731	0.4077
Fix Decoder	0.9699	0.5113	0.6209	0.6875
Fix Decoder + MLP Mid	0.9677	0.4998	0.6117	0.6547
Fix Decoder + MLP End	0.9731	0.5355	0.6582	0.6852
Fix Decoder + MLP All	0.9722	0.5396	0.6381	0.6643

Table 3: Evaluation Metrics with LoRA

Treatment	Test Token Accuracy	Top 1 Accuracy	Top 3 Accuracy	Top 5 Accuracy
LoRA All	0.9742	0.5319	0.6489	0.6786
LoRA All + MLP All	0.9720	0.4904	0.6037	0.6422
LoRA All + MLP End	0.9738	0.5357	0.6470	0.6801
LoRA All + MLP Mid	0.9715	0.4959	0.6104	0.6404
LoRA Encoder + Fixed Decoder	0.9686	0.5045	0.6154	0.6523
Fixed Encoder + LoRA Decoder	0.9752	0.5534	0.6584	0.6937
Fixed Encoder + LoRA Decoder + MLP All	0.9750	0.5461	0.6507	0.6834
Fixed Encoder + LoRA Decoder + MLP End	0.9741	0.5479	0.6626	0.6937
Fixed Encoder + LoRA Decoder + MLP Mid	0.9742	0.5352	0.6453	0.6760

5 Future Work

We plan to apply distillation between LlasMol model and our Bart-based LM in fine-tuning as we already observe that the LM generated output could help in producing a better fine-tuned model. Apart from the original dataset that the model is trained on, distillation also takes the behavior of LLM (LlasMol) into account when computing loss. This may help our model to be better fine-tuned. Further more, we may consider better decoding strategy since our model gives potentially higher token level accuracy but lower molecular level accuracy comparing to the baseline.

6 URL of our project repo and dataset repo

Project: `https://github.com/scaliaven/NLP_project`

Dataset(LLM generated along with original output): `https://huggingface.co/datasets/scaliaven/Ustop50k`

References

[1] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, 2019. PMID: 31572784.

[2] Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset, 2024.

[3] Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, jan 2022.

[4] Daniel Lowe. Chemical reactions from us patents (1976-sep2016), 2017. URL https://doi.org/10.6084/m9.figshare.5104873.v1.