Evaluating Chemistry Prompts for Large-Language Model Fine-Tuning

Carmelo Gonzales
Intel Labs

Michael Pieler
Stability AI

Kevin Jablonka
Laboratory of Organic and Macromolecular Chemistry (IOMC), FSU Jena, Germany
and
Helmholtz Institute for Polymers in Energy Applications Jena (HIPOLE Jena), Germany

Santiago Miret
Intel Labs

Abstract

We perform a study of large language model (LLM) templating and data presentation in the field of chemistry and materials science by analyzing memorization and generalization performance of a LlaMa model fine-tuned on 34 unique datasets. As application domains for LLMs become more specialized, it becomes more and more important to understand the impacts of training data, templates, and evaluations. While many pretrained LLMs have observed enormous corpora of text data, they are not guaranteed to be useful in domain specific tasks which may involve specialized data and prompts, such as chemistry and materials science. To further understand the capabilities of LLMs, we study the performance of various fine-tuned base models and show how differences in template styles with varying molecular string representations affect model performance. We hope that these insights may serve as a helpful path towards future larger scale training for chemistry and materials science specific LLMs.

1 Introduction

Recent breakthroughs in natural language processing (NLP) have opened up new possibilities for automating technical tasks using language models. Large language models (LLMs), in particular, have shown great promise in solving complex problems across various domains such as education, finance, and software development [1]. In the scientific domains of chemistry and material science, LLMs are being applied to problems in climate change, materials discovery, and property prediction [2–7]. While advancements have continued, the need for high-quality data and evaluation pipelines are needed, particularly when the data representations become specialized such that the model has never encountered them before [8, 9]. In these nascent application fields, understanding how input data representations and prompts are learned is an important consideration for domain researchers to develop practical tools that solve real-world science challenges [10–13].

2 Background

Recent work by Mirza et al. [10] provides a greater understanding of the capabilities of LLMs for the chemistry domain. Mirza et al. [10]'s analysis showed that larger models that are likely

3rd Workshop for AI Accelerated Materials Design—NeurIPS 2024, Vancouver

to have observed a greater amount of chemistry-related text in their training data perform better in many chemistry tasks. Mirza et al. [10], however, did not perform any domain-specific fine-tuning or detailed prompt evaluation on the benchmarked LLMs. Prior work has shown that LLM fine-tuning can imbue LLMs for domain-specific language that leads to better performance on domain-specific tasks [9, 14, 15]. On top of that, prompt engineering has shown a significant impact on modeling performance for diverse LLM applications, including chemistry and materials science [16, 2, 17, 9, 13].

3 Dataset

Building on top of chemical data from Mirza et al. [10], Xu et al. [18], OpenBioML [19], we construct various structured prompt templates based on tabular data that provide concrete answers. Our sampling pipeline consists of several steps: First, we create data samples from the diverse molecular text representations [20] (e.g., SMILES, canonical SMILES, DeepSMILES, SELFIES, InChI, and the IUPAC nomenclature) using common computational tools, such as RDKit [21]. Additionally, we split the dataset into train, validation, and test subsets based on the Murcko scaffolds of the molecules taking into account potentially pre-defined splits [22] . Next, we insert the sampled chemical structure into the defined placeholders of the prompt templates. The final templates then consist of the prompt template itself with all placeholders filled in, including the different molecular representations, as well as relevant answers based on multiple choice format or true/false statements. For the multiple-choice format, the enumeration symbol, i.e., lower (`abc...`) and upper case letters (`ABC...`) or numeric characters (`123...`), and additional wrapping characters, i.e., `.`, `.)`, `)`, `:`, `()`, `[]`, are sampled. The prompt templates used for fine-tuning are shown in Appendix A.1.

4 Experiments

To assess the ability of an LLM to learn from chemistry-related data, we perform two key experiments: first, to evaluate how effectively the LLM retains and memorizes its training data; second, to measure how well it can generalize and apply this learned knowledge to novel contexts and unseen data. To do this, we first present various prompting templates as well as different chemical representations to a model during fine-tuning, and then evaluate the fine-tuned model both on its own training data, as well as a holdout test dataset. By evaluating the model's ability to both memorize its own training data and generalize to unseen data, we aim to provide a deeper understanding of how models best learn and understand textual knowledge in the chemistry domain, thereby helping to set a path toward future LLM model training and deployment.

4.1 Templates

For the primary task, the main objective of the LLM is to determine whether a particular molecule has the characteristic of being mutagenic. The prompt templates used to construct datasets take a few different forms, which range in difficulty, and are showing in Appendix A.1. The most basic templates are simply a statement about whether a particular molecule exhibits mutagenic properties or not, for example, template 0. Other templates use a statement-response format with different forms of context and model expectations contained in the statement itself. The objective of these templates is to determine whether the statement is true or false, as seen in template 5. Finally, the majority of other templates take the form of question-answer prompts, with context and model expectations mixed in. All multiple-choice style templates have two options and one correct answer choice as an output, except for template 15 which may have multiple choices and multiple correct answers as an expected output.

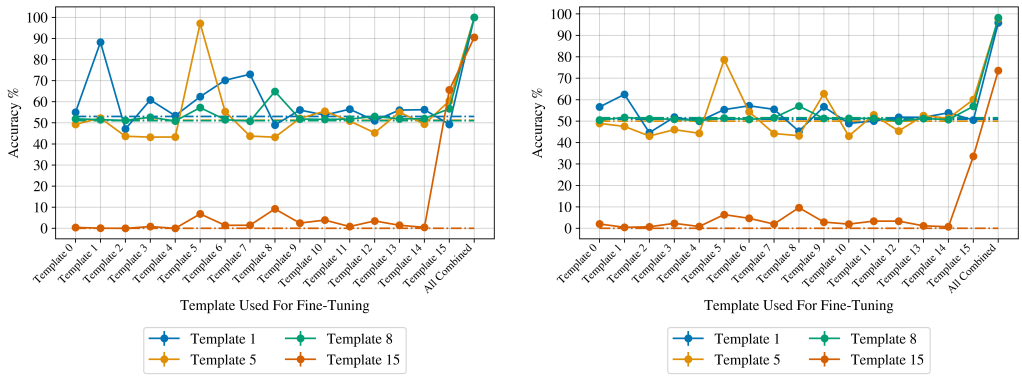
A set of 16 prompting templates are used to structure the training and test data, each of which relies on different methods of presenting the question, data, context, and expected model output. Out of the 16 templates, four are chosen to benchmark the fine-tuned models. In terms of benchmark difficulty, template 0 is considered "easy", templates 5 and 8 are considered "medium", and template 15 is considered "hard". On top of the predefined templates, a concatenation of all data from templates 0-15 is constructed to show how a model fine-tuned on all input data performs, and is called "All Combined".. The datasets are then prepared using two collections of molecule string representations. One set of experiments uses only the SMILES representation, while the second set uses a combination

of SMILES, Deep SMILES, canonical SMILES, SELFIES, InChI, and IUPAC molecular string conventions. The result of using 17 templates and two molecular representation collections is 34 total datasets used to independently fine-tune the base model.

4.2 Fine-Tuning with LLama2-7B

The templates and data representations are used to fine-tune the pretrained LLama2-7B [23] model using LoRA [24] with the default tokenizer. Model configuration and hyper-parameters for fine-tuning may be found in Appendix A.3. For each template, the base model instance is fine-tuned for 10 total epochs, resulting in 17 models for the two collections of data representations.

After all models have been fine-tuned on their respective templates, each model is evaluated using the four benchmarking templates using the model completions only. Note that in these evaluations, the benchmarking data is the same as the train data for templates 1, 5, 8, and 15. For models that have not been trained on a template that is also used as a benchmark, the context and question style have not been seen during training, however the specific molecular data strings are the same. The base rate of the unmodified original model is also calculated and compared to each fine-tuned model. The evaluation metric used is accuracy, in which the model output is compared directly to the expected output and scored as either 0 or 1 for 0% accurate or 100% accurate. In the template 15 benchmark, where more than one answer choice may be correct, the exact-match accuracy is used.



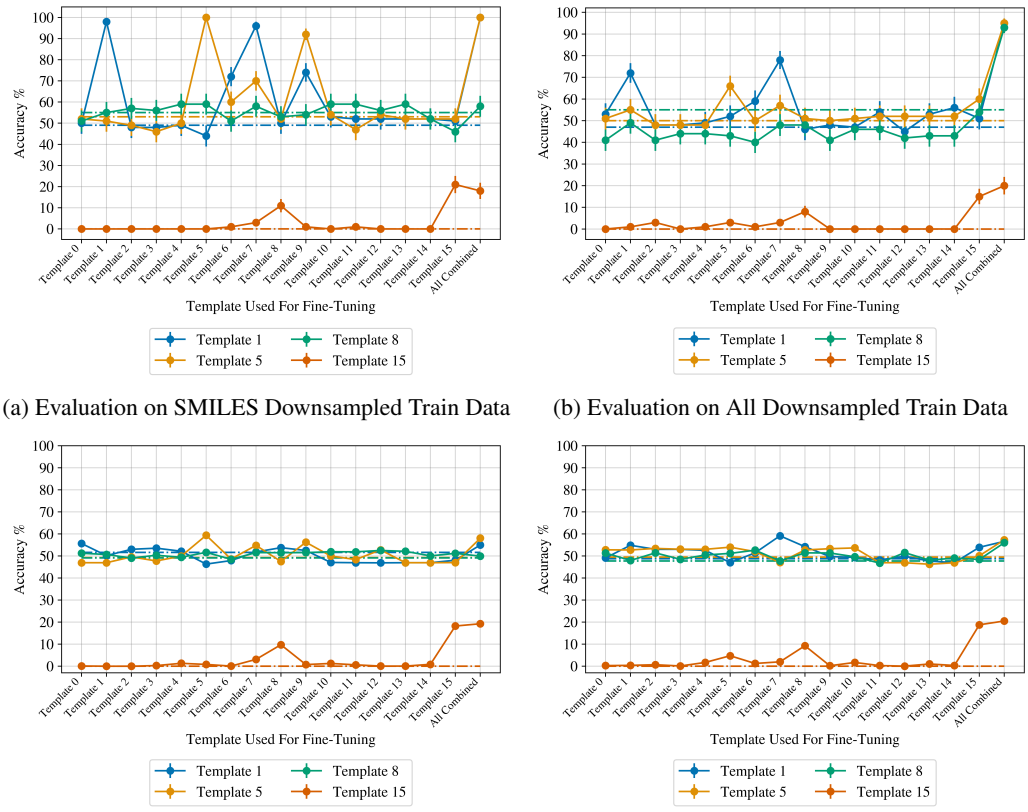
(a) Evaluation on SMILES Train Data

(b) Evaluation on All Train Data

Figure 1: Each fine-tuned model is evaluated against the four benchmarking templates for both of the data representation collections, as well as the base rate for each template (dashed lines). On the left, the SMILES only collection shows high memorization accuracy for templates 1 and 5 and 15 relative to the base rate, while the full data collection (right) shows high memorization accuracy for templates 5 and 15. In both data collections, the memorization rate for the model fine-tuned on the collection of templates (All Combined) shows high memorization accuracy.

First, looking at models trained on the same template in which they are evaluated on shows that models are memorizing some of the information which is presented, however the overall accuracy score is far lower than 100% meaning that even after 10 full epochs of training, the information is not fully retained by the model. Second, some templates seem to have a negative effect on model memorization, seen by accuracy scores falling below the base rate. Third, the model fine-tuned on the combination of all templates reaches near-perfect memorization on all templates, except for template 15, which does, however, still see a significant improvement from the base rate. These observations hold true across both collections of molecular representations, with the SMILES only string collection performing slightly better. The benchmark for template 15 is clearly much harder for both the base and fine-tuned models.

In addition to fine-tuning the full set of train data for 10 epochs, a reduced set of training data containing 100 total samples for each template is used for fine-tuning over 1,000 epochs to see if the models are able to fully memorize the training data, and if training longer may boost the overall performance of models trained on templates not in the benchmark set.



(a) Evaluation on SMILES Downsampled Train Data

(b) Evaluation on All Downsampled Train Data

(c) Evaluation on SMILES Downsampled Test Data

(d) Evaluation on All Downsampled Test Data

Figure 2: In 2a and 2b, the memorization accuracy is high for many templates, and some correlation can be seen by accuracies peaking for multiple fine-tuned models on the same benchmark. In 2c and 2d, performance on the holdout test set is poor across all models, with on only the template 15 and combined dataset fine-tuned models showing any significant improvement. This may be a result of severe over-fitting or simply not enough training data.

While training for more epochs on less data seems to help the performance of models evaluated against the templates in which they were trained on, there is not a significant boost to all other fine-tuned models with the exception of a few templates. In the SMILES representation collection, the correlation between some templates becomes easier to see. For example, models fine-tuned on template 1 and template 7 both achieve a high level of memorization when evaluated on template 1. Additionally, the models which were fine-tuned on the concatenation of all templates have near-perfect memorization, similar to the prior experiments. Interestingly, for both data representation collections, the model fine-tuned on template 8 and evaluated on its own training data does not show any sign of memorization. In the collection of all molecular string representations, all models evaluated on template 8 fall below the base rate, except the ”All Combined” model.

Overall, models evaluated against the template in which they were fine-tuned with will perform better than models fine-tuned with another template, regardless of their similarity. Additionally, models fine-tuned using samples from each template perform just as well, or better, than the models fine-tuned on individual templates. Due to these observations, simply using one template for training LLM’s in this domain may lead to results that may sometimes be worse than the starting base model.

5 Discussion

In the fields of chemistry and material science, LLMs show promise in complimenting molecular discovery, property prediction, and general educational workflows, however base foundation models may lack the expert level domain knowledge to be useful to their fullest extent. In this work, a

study on how LLMs learn from various domain specific prompts, contexts, and datasets is used to get a better understanding of what may or may not work when fine-tuning an off the shelf model. While memorization performance is high among many of the fine-tuned models, their generalization performance to unseen data is still lacking, leading to more questions about how best to structure and present chemistry datasets. Models fine-tuned on the combination of all templates show strong performance in memorization and better performance in generalization as compared to fine-tuned and base models, solidifying the general understanding that more and diverse data inputs lead to more performant models.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, and Berend Smit. Leveraging large language models for predictive chemistry. *Nature Machine Intelligence*, 6(2):161–169, 2024.

[3] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pages 1–11, 2024.

[4] Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

[5] Yu Song, Santiago Miret, and Bang Liu. Matsci-nlp: Evaluating scientific language models on materials science language tasks using text-to-schema modeling. *arXiv preprint arXiv:2305.08264*, 2023.

[6] Andrew D White. The future of chemistry is language. *Nature Reviews Chemistry*, 7(7): 457–458, 2023.

[7] Mara Schilling-Wilhelmi, Martiño Ríos-García, Sherjeel Shabih, María Victoria Gil, Santiago Miret, Christoph T Koch, José A Márquez, and Kevin Maik Jablonka. From text to insight: Large language models for materials science data extraction. *arXiv preprint arXiv:2407.16867*, 2024.

[8] Nawaf Alampara, Santiago Miret, and Kevin Maik Jablonka. Mattext: Do language models need more than text & scale for materials modeling? In *AI for Accelerated Materials Design-Vienna 2024*, 2024.

[9] Yu Song, Santiago Miret, Huan Zhang, and Bang Liu. Honeybee: Progressive instruction finetuning of large language models for materials science. In *AI for Accelerated Materials Design - NeurIPS 2023 Workshop*, 2023. URL https://openreview.net/forum?id=TLercqxR4f.

[10] Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Benedict Emoekabu, Aswanth Krishnan, Mara Wilhelmi, Macjonathan Okereke, Juliane Eberhardt, Amir Mohammad Elahi, Maximilian Greiner, et al. Are large language models superhuman chemists? *arXiv preprint arXiv:2404.01475*, 2024.

[11] Santiago Miret and NM Krishnan. Are llms ready for real-world materials discovery? *arXiv preprint arXiv:2402.05200*, 2024.

[12] Santiago Miret, NM Anoop Krishnan, Benjamin Sanchez-Lengeling, Marta Skreta, Vineeth Venugopal, and Jennifer N Wei. Perspective on ai for accelerated materials design at the ai4mat-2023 workshop at neurips 2023. *Digital Discovery*, 2024.

[13] Huan Zhang, Yu Song, Ziyu Hou, Santiago Miret, and Bang Liu. Honeycomb: A flexible llm-based agent system for materials science. *arXiv preprint arXiv:2409.00135*, 2024.

[14] Tong Xie, Yuwei Wan, Wei Huang, Zhenyu Yin, Yixuan Liu, Shaozhou Wang, Qingyuan Linghu, Chunyu Kit, Clara Grazian, Wenjie Zhang, et al. Darwin series: Domain specific large language models for natural science. *arXiv preprint arXiv:2308.13565*, 2023.

[15] Vaibhav Mishra, Somaditya Singh, Mohd Zaki, Hargun Singh Grover, Santiago Miret, Mausam ., and N M Anoop Krishnan. LLamat: Large language models for materials science. In *AI for Accelerated Materials Design - Vienna 2024*, 2024. URL https://openreview.net/forum?id=ZUkmRy6SqS.

[16] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

[17] Kevin Maik Jablonka, Qianxiang Ai, Alexander Al-Feghali, Shruti Badhwar, Joshua D Bocarsly, Andres M Bran, Stefan Bringuier, L Catherine Brinson, Kamal Choudhary, Defne Circi, et al. 14 examples of how llms can transform materials science and chemistry: a reflection on a large language model hackathon. *Digital Discovery*, 2(5):1233–1250, 2023.

[18] Congying Xu, Feixiong Cheng, Lei Chen, Zheng Du, Weihua Li, Guixia Liu, Philip W Lee, and Yun Tang. In silico prediction of chemical ames mutagenicity. *Journal of chemical information and modeling*, 52(11):2840–2847, 2012.

[19] OpenBioML. Chemnlp: Language models for chemistry. https://github.com/OpenBioML/chemnlp, 2023. Accessed: 2024-09-05.

[20] Mario Krenn, Qianxiang Ai, Senja Barthel, Nessa Carson, Angelo Frei, Nathan C Frey, Pascal Friederich, Théophile Gaudin, Alberto Alexander Gayle, Kevin Maik Jablonka, et al. Selfies and the future of molecular string representations. *Patterns*, 3(10), 2022.

[21] Greg Landrum. Rdkit documentation. *Release*, 1(1-79):4, 2013.

[22] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[24] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

A Appendix

A.1 Templates

Each of the 16 templates used for fine-tuning and benchmarking is summarized below.

A.2 Figure Data

The data from each figure is presented in the following tables.

A.3 Fine-Tuning Hyper-Parameters

For all experiments, the same set of hyper-parameters is used with the exception of number of epochs, where the first set of experiments uses 10, and the second set of experiments uses 1,000.

Table 1: Templates arranged by number, and indicating if the particular template was used for benchmarking or not.

Template Number	Template	Benchmark?
0	The {#molecule |!}{SMILES_description} {SMILES#} is {mutagenic#not &NULL}{mutagenic_names_adjective}.	
1	Is the {SMILES_description} {SMILES#} {mutagenic_names_adjective}: {mutagenic#no&yes}	Yes
2	The molecule with the {SMILES_description} {#representation of |!}{SMILES#} {#shows|exhibits|displays!} {mutagenic#no &NULL}{mutagenic_names_adjective} properties.	
3	Based on the {SMILES_description} {#representation |!}{SMILES#}, the molecule has {mutagenic#no &NULL}{mutagenic_names_adjective} {#properties|characteristics|features!}.	
4	The {SMILES_description} {SMILES#} {#represents|is from!} a molecule that is {mutagenic#not &NULL}identified as {mutagenic_names_adjective}.	
5	Task: Please classify a molecule based on the description. Description: A molecule that is {mutagenic_names_adjective}. {#Molecule |!}{SMILES_description}: {SMILES#} Constraint: Even if you are {#uncertain|not sure!}, you must pick either "True" or "False" without using any {#other|additional!} words. Result: {mutagenic#False&True}	Yes
6	User: Can you {#tell me|derive|estimate!} if the molecule with the {SMILES_description} {SMILES#} is {mutagenic_names_adjective}? Assistant: {mutagenic#No&Yes}, this molecule is {mutagenic#not &NULL}{mutagenic_names_adjective}.	
7	User: Is the molecule with the {SMILES_description} {SMILES#} {mutagenic_names_adjective}? Assistant: {mutagenic#No&Yes}, it is {mutagenic#not &NULL}{mutagenic_names_adjective}.	
8	Task: Please answer the multiple choice question. Question: Is the molecule with the {SMILES_description} {#representation of |!}{SMILES#} {mutagenic_names_adjective}? Constraint: Even if you are {#uncertain|not sure!}, you must pick either {%multiple_choice_enum%2%aA1} without using any {#other|additional!} words. Options: {mutagenic%} Answer: {%multiple_choice_result}	Yes
9	Task: Please classify a molecule based on the description. Description: A molecule that is {mutagenic_names_adjective}. {#Molecule |!}{SMILES_description}: {SMILES#} Constraint: Answer the question in a {#full|complete!} sentence. Result: This molecule is {mutagenic#not &NULL}{mutagenic_names_adjective}.	
10	Task: Please {#give me|create|generate!} a {#molecule |!}{SMILES_description} based on the {#text |!}description{# below|!}. Description: A molecule that is {mutagenic#not &NULL}{mutagenic_names_adjective}. Result: {SMILES#}	
11	User: Can you {#give me|create|generate!} the {SMILES_description} of a molecule that is {mutagenic#not &NULL}{mutagenic_names_adjective}? Assistant: {#Yes|Of course|Sure|Yes, I'm happy to help!}, here you go: {SMILES#}	
12	User: I'm {#searching|looking!} for the {SMILES_description} of a molecule that is {mutagenic#not &NULL}{mutagenic_names_adjective}? Assistant: This is a molecule that is {mutagenic#not &NULL}{mutagenic_names_adjective}: {SMILES#}	
13	User: I want to {#come up with|create|generate!} a {#molecule |!}{SMILES_description}. Assistant: {#This sounds very exciting. |This sounds very interesting. !}Should I consider any {#constraints|specific points!} for the {#generation|creation!}? User: Yes, please. The molecule should {mutagenic#not &NULL}be {mutagenic_names_adjective}. Assistant: {#Ok|Got it!},{# here you go,|!} this {SMILES_description} is {mutagenic#not &NULL}{mutagenic_names_adjective}: {SMILES#}	
14	User: I want to {#come up with|create|generate!} a {#molecule |!}{SMILES_description}. Assistant: {#This sounds very exciting. |This sounds very interesting. !}Should it be a special {#molecule|one!}? User: Yes, the molecule should {mutagenic#not &NULL}be {mutagenic_names_adjective}. Assistant: {#Understood|Got it|Ok!}, this {SMILES_description} is {mutagenic#not &NULL}{mutagenic_names_adjective}: {SMILES#}	
15	Task: Please answer the multiple choice question. Question: Which molecules are {mutagenic#not &NULL}{mutagenic_names_adjective}? Constraint: You must select none, one or more options from {%multiple_choice_enum%2-5%aA1} without using any {#other|additional!} words. Options: {SMILES%mutagenic%} Answer: {%multiple_choice_result}	Yes

Table 2: Data for figure 1a.

Benchmark Template	Base Rate	Template 0	Template 1	Template 2	Template 3	Template 4	Template 5	Template 6	Template 7	Template 8	Template 9	Template 10	Template 11	Template 12	Template 13	Template 14	Template 15	All Combined
Template 1	53.045 ± 0.72	55.041 ± 0.717	88.215 ± 0.465	47.08 ± 0.72	60.861 ± 0.704	53.378 ± 0.719	62.399 ± 0.698	70.193 ± 0.66	73.02 ± 0.64	48.971 ± 0.721	56.08 ± 0.716	53.669 ± 0.719	56.475 ± 0.715	51.07 ± 0.721	56.059 ± 0.716	56.225 ± 0.715	49.304 ± 0.721	99.938 ± 0.036
Template 5	51.299 ± 0.721	49.283 ± 0.721	52.255 ± 0.72	43.65 ± 0.715	43.234 ± 0.714	43.297 ± 0.714	97.152 ± 0.24	55.352 ± 0.717	43.733 ± 0.715	43.193 ± 0.714	52.11 ± 0.72	55.519 ± 0.717	50.904 ± 0.721	45.271 ± 0.718	55.041 ± 0.717	49.387 ± 0.721	60.424 ± 0.705	100.0 ± 0.0
Template 8	51.029 ± 0.721	51.902 ± 0.72	51.507 ± 0.721	51.174 ± 0.721	52.65 ± 0.72	50.883 ± 0.721	57.223 ± 0.713	51.507 ± 0.721	50.821 ± 0.721	64.872 ± 0.688	51.819 ± 0.72	51.652 ± 0.721	51.777 ± 0.72	53.024 ± 0.72	51.881 ± 0.72	52.006 ± 0.72	56.724 ± 0.714	100.0 ± 0.0
Template 15	0 ± 0	0.436 ± 0.095	0.083 ± 0.042	0.0 ± 0.0	0.852 ± 0.133	0.0 ± 0.0	6.838 ± 0.364	1.434 ± 0.171	1.476 ± 0.174	9.208 ± 0.417	2.494 ± 0.225	3.908 ± 0.279	0.852 ± 0.133	3.513 ± 0.265	1.455 ± 0.173	0.478 ± 0.099	65.6 ± 0.685	90.543 ± 0.422

Table 3: Data for figure 1b.

Benchmark Template	Base Rate	Template 0	Template 1	Template 2	Template 3	Template 4	Template 5	Template 6	Template 7	Template 8	Template 9	Template 10	Template 11	Template 12	Template 13	Template 14	Template 15	All Combined
Template 1	50.904 ± 0.721	56.62 ± 0.715	62.461 ± 0.698	44.627 ± 0.717	51.86 ± 0.72	49.969 ± 0.721	55.373 ± 0.717	57.161 ± 0.714	55.498 ± 0.717	45.271 ± 0.718	56.724 ± 0.714	48.95 ± 0.721	50.031 ± 0.721	51.84 ± 0.72	51.798 ± 0.72	53.877 ± 0.719	50.53 ± 0.721	95.843 ± 0.288
Template 5	50.01 ± 0.721	48.95 ± 0.721	47.537 ± 0.72	43.047 ± 0.714	46.04 ± 0.719	44.336 ± 0.716	78.674 ± 0.591	54.355 ± 0.718	44.232 ± 0.716	43.193 ± 0.714	62.794 ± 0.697	43.047 ± 0.714	52.92 ± 0.72	45.396 ± 0.718	52.567 ± 0.72	51.341 ± 0.721	60.05 ± 0.706	97.818 ± 0.211
Template 8	51.549 ± 0.721	50.509 ± 0.721	51.694 ± 0.721	51.05 ± 0.721	50.925 ± 0.721	50.281 ± 0.721	51.465 ± 0.721	50.8 ± 0.721	51.59 ± 0.721	56.994 ± 0.714	51.299 ± 0.721	51.258 ± 0.721	51.299 ± 0.721	49.99 ± 0.721	51.216 ± 0.721	50.717 ± 0.721	56.911 ± 0.714	98.212 ± 0.191
Template 15	0 ± 0	2.057 ± 0.204	0.478 ± 0.099	0.727 ± 0.123	2.328 ± 0.217	0.852 ± 0.133	6.402 ± 0.353	4.677 ± 0.304	1.995 ± 0.202	9.645 ± 0.426	2.931 ± 0.243	1.975 ± 0.201	3.367 ± 0.26	3.367 ± 0.26	1.164 ± 0.155	0.748 ± 0.124	33.569 ± 0.681	73.561 ± 0.636

Table 4: Data for figure 2a.

Benchmark Template	Base Rate	Template 0	Template 1	Template 2	Template 3	Template 4	Template 5	Template 6	Template 7	Template 8	Template 9	Template 10	Template 11	Template 12	Template 13	Template 14	Template 15	All Combined
Template 1	49.0 ± 5.024	50.0 ± 5.025	98.0 ± 1.407	48.0 ± 5.021	48.0 ± 5.021	49.0 ± 5.024	44.0 ± 4.989	72.0 ± 4.513	96.0 ± 1.969	50.0 ± 5.025	74.0 ± 4.408	53.0 ± 5.016	52.0 ± 5.021	52.0 ± 5.021	52.0 ± 5.021	52.0 ± 5.021	51.0 ± 5.024	100.0 ± 0.0
Template 5	53.0 ± 5.016	52.0 ± 5.021	51.0 ± 5.024	49.0 ± 5.024	46.0 ± 5.009	50.0 ± 5.025	100.0 ± 0.0	60.0 ± 4.924	70.0 ± 4.606	52.0 ± 5.021	92.0 ± 2.727	54.0 ± 5.009	47.0 ± 5.016	54.0 ± 5.009	52.0 ± 5.021	52.0 ± 5.021	52.0 ± 5.021	100.0 ± 0.0
Template 8	55.0 ± 5.0	51.0 ± 5.024	55.0 ± 5.0	57.0 ± 4.976	56.0 ± 4.989	59.0 ± 4.943	59.0 ± 4.943	51.0 ± 5.024	58.0 ± 4.96	53.0 ± 5.016	54.0 ± 5.009	59.0 ± 4.943	59.0 ± 4.943	56.0 ± 4.989	59.0 ± 4.943	52.0 ± 5.021	46.0 ± 5.009	58.0 ± 4.96
Template 15	0 ± 0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	1.0 ± 1.0	3.0 ± 1.714	11.0 ± 3.145	1.0 ± 1.0	0.0 ± 0.0	1.0 ± 1.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	21.0 ± 4.094	18.0 ± 3.861

Table 5: Data for figure 2b.

Benchmark Template	Base Rate	Template 0	Template 1	Template 2	Template 3	Template 4	Template 5	Template 6	Template 7	Template 8	Template 9	Template 10	Template 11	Template 12	Template 13	Template 14	Template 15	All Combined
Template 1	47.0 ± 5.016	53.0 ± 5.016	72.0 ± 4.513	48.0 ± 5.021	48.0 ± 5.021	49.0 ± 5.024	52.0 ± 5.021	59.0 ± 4.943	78.0 ± 4.163	46.0 ± 5.009	48.0 ± 5.021	47.0 ± 5.016	54.0 ± 5.009	45.0 ± 5.0	53.0 ± 5.016	56.0 ± 4.989	51.0 ± 5.024	95.0 ± 2.19
Template 5	50.0 ± 5.025	51.0 ± 5.024	55.0 ± 5.0	48.0 ± 5.021	48.0 ± 5.021	48.0 ± 5.021	66.0 ± 4.761	50.0 ± 5.025	57.0 ± 4.976	51.0 ± 5.024	50.0 ± 5.025	51.0 ± 5.024	52.0 ± 5.021	52.0 ± 5.021	52.0 ± 5.021	52.0 ± 5.021	60.0 ± 4.924	95.0 ± 2.19
Template 8	55.0 ± 5.0	41.0 ± 4.943	49.0 ± 5.024	41.0 ± 4.943	44.0 ± 4.989	44.0 ± 4.989	43.0 ± 4.976	40.0 ± 4.924	48.0 ± 5.021	48.0 ± 5.021	41.0 ± 4.943	46.0 ± 5.009	46.0 ± 5.009	42.0 ± 4.96	43.0 ± 4.976	43.0 ± 4.976	54.0 ± 5.009	93.0 ± 2.564
Template 15	0 ± 0	0.0 ± 0.0	1.0 ± 1.0	3.0 ± 1.714	0.0 ± 0.0	1.0 ± 1.0	3.0 ± 1.714	1.0 ± 1.0	3.0 ± 1.714	8.0 ± 2.727	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	15.0 ± 3.589	20.0 ± 4.02

Table 6: Data for figure 2c.

Benchmark Template	Base Rate	Template 0	Template 1	Template 2	Template 3	Template 4	Template 5	Template 6	Template 7	Template 8	Template 9	Template 10	Template 11	Template 12	Template 13	Template 14	Template 15	All Combined
Template 1	51.606 ± 1.197	55.677 ± 1.19	50.172 ± 1.198	53.039 ± 1.195	53.555 ± 1.195	52.179 ± 1.196	46.33 ± 1.194	47.936 ± 1.197	51.835 ± 1.197	53.784 ± 1.194	52.523 ± 1.196	47.076 ± 1.196	46.961 ± 1.195	46.904 ± 1.195	46.961 ± 1.195	46.961 ± 1.195	48.108 ± 1.197	54.989 ± 1.192
Template 5	49.312 ± 1.198	46.961 ± 1.195	46.961 ± 1.195	49.599 ± 1.198	47.764 ± 1.196	49.943 ± 1.198	59.404 ± 1.176	48.624 ± 1.197	54.817 ± 1.192	47.534 ± 1.196	56.25 ± 1.188	50.0 ± 1.198	48.394 ± 1.197	52.752 ± 1.196	46.961 ± 1.195	46.961 ± 1.195	47.018 ± 1.195	58.085 ± 1.182
Template 8	49.14 ± 1.197	51.261 ± 1.197	50.688 ± 1.198	49.083 ± 1.197	50.344 ± 1.198	49.427 ± 1.198	51.663 ± 1.197	48.509 ± 1.197	51.548 ± 1.197	51.491 ± 1.197	51.548 ± 1.197	51.95 ± 1.197	51.892 ± 1.197	52.523 ± 1.196	52.122 ± 1.197	50.057 ± 1.198	51.204 ± 1.197	49.943 ± 1.198
Template 15	0 ± 0	0.115 ± 0.081	0.0 ± 0.0	0.0 ± 0.0	0.287 ± 0.128	1.319 ± 0.273	0.803 ± 0.214	0.057 ± 0.057	3.096 ± 0.415	9.748 ± 0.71	0.745 ± 0.206	1.261 ± 0.267	0.631 ± 0.19	0.057 ± 0.057	0.057 ± 0.057	0.86 ± 0.221	18.234 ± 0.925	19.266 ± 0.945

Table 7: Data for figure 2d.

Benchmark Template	Base Rate	Template 0	Template 1	Template 2	Template 3	Template 4	Template 5	Template 6	Template 7	Template 8	Template 9	Template 10	Template 11	Template 12	Template 13	Template 14	Template 15	All Combined
Template 1	48.796 ± 1.197	49.14 ± 1.197	54.874 ± 1.192	53.039 ± 1.195	53.039 ± 1.195	52.58 ± 1.196	47.018 ± 1.195	51.433 ± 1.197	59.174 ± 1.177	54.243 ± 1.193	49.771 ± 1.198	49.369 ± 1.198	48.222 ± 1.197	49.885 ± 1.198	47.706 ± 1.196	47.018 ± 1.195	53.956 ± 1.194	56.651 ± 1.187
Template 5	49.484 ± 1.198	52.81 ± 1.196	52.752 ± 1.196	53.44 ± 1.195	53.096 ± 1.195	53.039 ± 1.195	54.014 ± 1.194	52.007 ± 1.197	47.076 ± 1.196	52.752 ± 1.196	53.326 ± 1.195	53.67 ± 1.194	46.961 ± 1.195	46.961 ± 1.195	46.273 ± 1.194	46.961 ± 1.195	50.115 ± 1.198	57.339 ± 1.185
Template 8	47.764 ± 1.196	51.376 ± 1.197	47.993 ± 1.197	51.433 ± 1.197	48.509 ± 1.197	50.459 ± 1.198	51.204 ± 1.197	52.638 ± 1.196	47.764 ± 1.196	51.433 ± 1.197	51.376 ± 1.197	49.656 ± 1.198	46.789 ± 1.195	51.548 ± 1.197	48.108 ± 1.197	49.083 ± 1.197	48.452 ± 1.197	56.021 ± 1.189
Template 15	0 ± 0	0.287 ± 0.128	0.401 ± 0.151	0.688 ± 0.198	0.115 ± 0.081	1.72 ± 0.311	4.759 ± 0.51	1.261 ± 0.267	2.007 ± 0.336	9.289 ± 0.695	0.229 ± 0.115	1.72 ± 0.311	0.287 ± 0.128	0.057 ± 0.057	1.032 ± 0.242	0.344 ± 0.14	18.807 ± 0.936	20.528 ± 0.967

Table 8: LoRA Parameters.

Alpha	8
Dropout	0.1
r	16

Table 9: Trainer Hyper-Parameters.

Batch Size	2
Learning Rate	2e-4
Weight Decay	0.001
Max Grad Norm	0.3
Warmup Ratio	0.3
LR Scheduler	Linear