Exploring the Recall of Language Models: Case Study on Molecules

Anonymous ACL submission

1 Introduction

Evaluating the performance of generative models, particularly large language models (LLMs), is an important challenge in modern deep learning (Chang et al., 2024). One overlooked aspect of evaluation is models' ability to generate *all* correct outputs for a given input. In scientific discovery, generation of new molecules with given characteristics is a cornerstone problem. For example, in drug discovery, most of the generated molecules may prove to be useless in the subsequent stages of drug development, so generating a diverse and ideally complete set of initial molecules is useful. To the best of our knowledge, the ability of LLMs to cover all correct outputs has not been systematically evaluated. There are two significant obstacles. First, it is hard to come up with a benchmark that lists all correct outputs. Second, the representations of the objects we are trying to generate are not often unique. In this paper we propose a benchmark that overcomes both obstacles and enables research on optimizing recall of LLMs.

2 Problem Definition

Let S be the set of all correct generations, i.e. strings. Assume there is an equivalence relation among the strings in S which divides S into M equivalence classes. We denote the set of unique equivalence classes by S^u. Each equivalence class corresponds to an object. For any object $m \in S^u$, the number of distinct strings corresponding to that object, is denoted by $\|m\|$.

The goal is to train a model that is able to generate from a maximum number of equivalence classes, i.e. unique objects. To achieve that, we train an LLM on a subset of M objects and evalute on subset of V objects (V < M) distinct from M. After training we generate G number of strings by sampling from the model. True positives, denoted by TP, are the generated strings that belong to S. Note, G can contain both duplicate strings and distinct strings that belong to the same equivalence class. Hence we also define *unique true positives*, $TP^u = |G^u|$, as the number of equivalent classes represented in G. We track two metrics:

$$Precision(G) = \frac{TP}{G} \ , \ Recall(G) = \frac{TP^u}{M} \quad (1)$$

If G is sampled in an i.i.d. fashion, TP scales linearly with G, and precision will not depend on G (after sufficiently large number of generations). On the contrary, TP^u does not scale indefinitely with G as it is upper bounded by $M = |S^u|$. Hence, the recall increases with G, can reach M and remain constant. The ideal model can learn to put uniform $p = \frac{1}{M}$ probability on all objects of the set S^u. Note that in this ideal scenario, the probability of each object can be distributed over its string representations in an arbitrary way. The recall of the ideal model after G generations will be $1 - (1-p)^G$. This serves as an upper bound for i.i.d. sampling methods.

2.1 Molecular Datasets

GDB-13 (Blum and Reymond, 2009) is an exhaustive set of molecules with at most 13 heavy atoms that satisfy certain conditions. We define the similarity $sim(m_1, m_2)$, between molecules, as the Tanimoto similarity (Tanimoto, 1958) between their MACCS fingerprints (Durant et al., 2002). Next, we define three subsets of GDB-13.

S_{asp} is the set of (all strings of) molecules from GDB-13 that have at least 0.4 similarity with *aspirin*. $S_{d>p}$ is the set of molecules that have at least 0.4 similarity to paracetamol (a famous drug), and have less than 0.4 similarity to 4-nitroanisole (a famous toxic molecule). $S_{d=p}$ is the set of molecules m that are at a similar distance from paracetamol (d) and 4-nitroanisole (p): $0.2 \leq sim(m, d) \leq 0.2165$ and $0.2 \leq sim(m, p) \leq 0.2165$.

Note that similarity in terms of MACCS fingerprints implies shared substructures between molecules. Hence, S_{asp} contains molecules that

		Precision (%)			**Recall (%)**		
Pretraining	**Fine-tuning**	S_{asp}	$S_{d>p}$	$S_{d=p}$	S_{asp}	$S_{d>p}$	$S_{d=p}$
Canonical	Canonical	75.69	68.27	14.07	55.02	46.83	15.19
Canonical	Randomized	70.59	61.63	10.86	53.74	44.90	12.39
Randomized	Canonical	**76.16**	**68.93**	**14.58**	54.80	46.76	**15.67**
Randomized	Randomized	75.15	65.66	13.67	**56.40**	**47.48**	15.33
Upper bound (i.i.d.)		100	100	100	70.09	65.76	71.12

Table 1: **Precision** and **Recall** of OPT-1.3B models fine-tuned on three sets of molecules, evaluated on 10 million strings generated with random sampling.

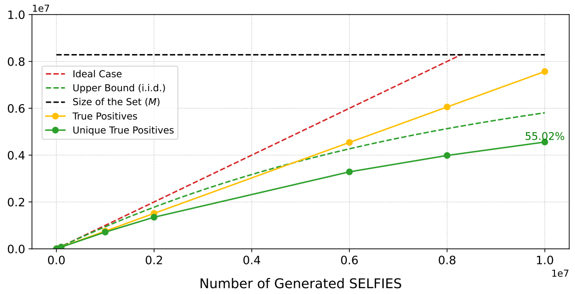

Figure 1: Number of TP and TP^u molecules generated by LLM fine-tuned on S_{asp}.

share some substructures with aspirin. $S_{d>p}$ is a more complex set as it contains molecules that share some substructures with paracetamol, but also do not share many structures with a toxic substance. We represent molecules with canonical and randomized SELFIES (Krenn et al., 2020) which are defined as the SELFIES of canonical and randomized SMILES produced by the RDKit library.

3 Experiments

We pretrain on a large subset of GDB-13, that excludes the three sets defined above. This data is split into a training set and a 10,000-instance validation set. We then finetune the LLMs on the three sets, using canonical and randomized SELFIES. From each set, we randomly select 1 million instances for training and 500 instances for evaluation. We adopt the majority of the pretraining settings and model architecture from OPT 1.3B (Zhang et al., 2022). We train from scratch for one epoch. For tokenization, we use an off-the-shelf tokenizer from (ZJUNLP, 2024).

4 Results

We used random sampling with temperature 1.0 to generate 10 million molecules from each of the models with results displayed in Table 1. As expected, S_{asp} is the easiest set, followed by $S_{d>p}$ and then by $S_{d=p}$. In contrast with the findings of (Arús-Pous et al., 2019), there is a little difference between the models trained on randomized and canonical SELFIES. For precision, fine-tuning on canonical is better, and for recall, fine-tuning on randomized is preferable.

Figure 1 shows how TP and TP^u grow as the number of generated strings grows. The plot indicates that the recall is close to saturation at 10 million generations, which motivates other approaches to improve LLM recall.

4.1 Predicting recall without generating

Here, we show that it is possible to predict the recall after G generated samples without actually generating them. We compute the probability that a molecule from the validation set will be generated in G attempts. Using a model, we can compute the expected probability of an entire string to be generated. Let $p_{i,j}$ denote the probability of generating the j-th string of the i-th molecule m_i. The average probability of a correct molecule generation in one attempt becomes: $\sum_{i=1}^{M} \sum_{j=1}^{\|m_i\|} p_{i,j}$. This is the expected precision of the model.

To estimate recall for G sampling iterations, we take the probability that the i-th given molecule will *not* be sampled in G iterations, and subtract it from one: $1 - \left(1 - \sum_{j=1}^{\|m_i\|} p_{i,j}\right)^G$. The expected value of this quantity over all molecules is the expected recall at G generations. Assuming access to a small validation set of V molecules, one can estimate the precision and recall using:

$$Precision = \frac{M}{V} \sum_{i=1}^{V} \sum_{j=1}^{\|m_i\|} p_{i,j}, \quad (2)$$

$$Recall = \frac{1}{V} \sum_{i=1}^{V} \left(1 - \left(1 - \sum_{j=1}^{\|m_i\|} p_{i,j} \right)^G \right) \quad (3)$$

We estimate precision and recall for various combinations of pretraining and finetuning, and molecular sets. The Pearson correlation between predicted and actual values for precision and recall are 0.99975 and 0.99982, respectively.

References

Josep Arús-Pous, Simon Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. 2019. Randomized smiles strings improve the quality of molecular generative models. *Journal of Cheminformatics*, 11.

Lorenz C Blum and Jean-Louis Reymond. 2009. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. *Journal of the American Chemical Society*, 131(25):8732–8733.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. 2002. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. 2020.

Taffee T Tanimoto. 1958. Elementary mathematical theory of classification and prediction.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *ArXiv*, abs/2205.01068.

ZJUNLP. 2024. Molgen-large. `https://huggingface.co/zjunlp/MolGen-large`. Accessed: 2024-05-22.