How to decode the effectiveness of a decoder

Gabriel Marcus
University of Cape Town
Cape Town, South Africa
mrcgab004@myuct.ac.za

ABSTRACT

Drug discovery is a critical industry. Molecular design and discovery are a vital role to enable drug discovery. New diverse Molecules have been discovered through the use of computational models. Molecular string representations have had large improvements recently. Namely SELFIE and SAFE. These molecular string representations, which have great improvements over the original SMILES can be used to improve AI-driven molecular design. Through high quality AI models, drug discovery can be sped up allowing for life-saving drugs to be produced.Generative-pre trained models can be improved through the selection of the correct decoding technique. A strong decoding algorithm can output more valid, diverse and unique molecules. Therefore SAFE-GPT can be improved through better inference decoding. Exploring different decoding techniques and classifying the objective of SAFE-GPT allows one to choose the correct decoding technique to improve its output. Through this improved output more diverse, drug-like molecules can be discovered and used towards the creation of new improved drugs enabling lives to be improved and saved.

INTRODUCTION

Drug-discovery is a long and arduous process. Machine learning can be used speed-up the process of discovering diverse drug-like molecules enabling the creation of new drugs. Natural language processing has had a significant impact in this field recently with the creation of the transformer model. [15]

The use of string based molecular representation has seen an improvement in molecular generation recently.[6, 10] With aid from the transformer model[10], the task of molecular generation can be further developed and improved from the current industry standard.[10, 16] This would lead to a large increase in the discovery of viable molecular structures and would allow for faster drug development, helping the entire population.

Generative pre-trained transformer (GPT) models are effective but can be optimised through improved inference-decoding.[14] One always needs a decoding algorithm to generate from a language model but both greedy decoding and pure sampling have issues.

In exploring decoding techniques, one must categorise the type of problem being solved, is the task, text-generation, summarisation which are both considered directed or is it story generation which is considered open-ended generation. These different categorisations will require different types of decoding techniques, namely deterministic or stochastic.[5, 8, 14]

One can also restrict the decoders output to reduce invalid molecules produced. This would be done by constraining the decoder. Constraining the model is a method of adding additional knowledge to the model.[3]

MOLECULAR REPRESENTATIONS

Discovering new molecules brings large societal progress, aiding in curing diseases. The set of the viable molecules which can be used in the creation of chemicals is computationally intractable. Estimates of the number of pharmacologically valid molecules is in the range of 10^{23} to 10^{80} compounds.[11]

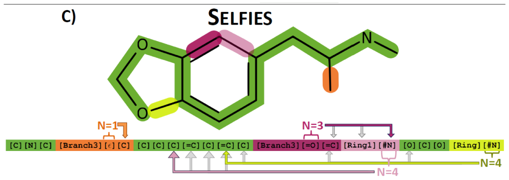
Many various approaches in searching the chemical space both in silico and in vitro have been conducted, including combinatorial libraries, evolutionary and high throughput screening. Current works shows that machine learning can utilised in the production process of molecules.[11] There have been several text-representations of molecules. The first widely adopted notation was SMILES, (Simplified Molecular Input Entry System).[16] SMILES was based on the principles of molecular graph theory and was first published in 1987.[16] The ease of use by chemist and machine compatibility allowed for chemical computer applications to be created.[16] SMILES is still seen as the standard for string-based representation of molecular information.[6] SMILES has a non-sequential depiction of molecular substructures.[10] SMILES notational structure allows for invalid molecules.[6] Therefore a SMILES representation of a molecule could be syntactically invalid or not follow fundamental chemical rules.[6] SMILES is unable to produce molecular substructures which have contiguous representations.[10] The result of SMILES weaknesses has spurred other representation to be derived. These other representations were introduced to improve on SMILES shortcomings, namely SELFIES (SELF-referencing Embedded Strings) and SAFE (Sequential Attachment based Fragment Embedding).

The SELFIE notational structure always guarantees a valid molecule. [6] SELFIEs are syntactically safe, as even a random SELFIE string will be a valid molecule. [6] In generative tasks SELFIEs are shown to outperform SMILES and produced a notably higher diversity of molecules. Producing highly diverse molecules is the primary goal in inverse design.[6]

A SELFIE string structure is modular. The ring size and branch length are stored in their identifiers, Ring and Branch respectively. The symbol following Ring and Branch represents the lengths. SELFIE symbols are generated using derivation rules from its own data table. [6]

Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) are made use of in showing the effectiveness of SELFIES. [6]

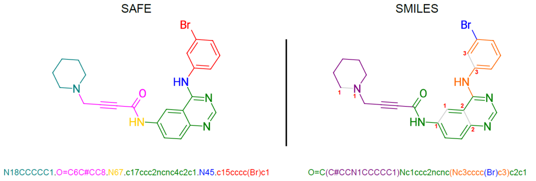
Variational Autoencoders use an architecture consisting of an encoder-decoder network. VAEs make use of stochastic decoding techniques where it forms its predictions by sampling laten space representations from a learned probability distribution. [6] Generative Adversarial networks do not make use of a decoder in their architecture. Instead GANs make use of a generator and a discriminator.


C)
SELFIES
O
O
N
N=1
N=3
[C] [N] [C] [Branch3] [ϵ] [C] [C] [C] [C] [=C] [C] [=C] [C] [Branch3] [=O] [=C] [Ring1] [#N] [O] [C] [O] [Ring] [#N]
N=4
N=4

[6]

Figure 1: Image of molecule MDMA with SELFIE string representation

SAFE notation was introduced to address the issue of the non-sequential depiction of molecular substructures that SMILES possessed. [6] The non-sequential depiction of molecular substructures design of SMILES makes it challenging for AI-driven molecular design.[10] SAFE represents molecules as an unordered sequence of fragment blocks. SAFE is a molecular notation which re-invents SMILES as a group of connected fragments
SAFEs design is compatibility with existing SMILES Parsers. [10] SAFEs design removes the need to recreate/modify existing applications.
SAFE notational structure contributed to large improvements in molecular generation with the aid of its research specialised SAFE-GPT. The researchers trained a decoder-only model on the safe notation and produced desirable results.


SAFE
SMILES
Br
HN
N
N
O
HN
N
1
2
3
N18CCCCC1.O=C6C#CC8.N67.c17ccc2ncnc4c2c1.N45.c15cccc(Br)c1
O=C(C#CCN1CCCCC1)Nc1ccc2ncnc(Nc3cccc(Br)c3)c2c1

[10]

Figure 2: The same molecule in both SAFE and SMILES notation

It is clear that the SAFE notation is easier to read in comparison to the SMILES notation. The same is also true for the SELFIE notation. These ease of makes the lives of chemist and researches easier.

The SAFE-GPT paper never mentions the decoder it makes use of. However it is posited that the model can be improved with a better decoding algorithm, which would address the shortcomings of the model they created.It is mentioned that the default settings were used.So it is likely they made use of a simple decoding algorithm. [10]

MOLECULAR EVALUATION

To determine the validity of molecules produced and thus the effectiveness of the model one can make use of the Python libary RDKit. This is an -out the box toolkit enabling effective evaluation. One can measure the Validity of the molecule. The validity of the molecules is the percentage of chemically valid structures.
The Uniqueness of a molecule is the fraction of non-duplicate molecules. It is determining if molecules are distinct from one-another.
The Diversity assesses the structural differences between the molecules. It is evaluating the internal diversity of generated molecules using the average pairwise Tanimoto distance(ECFP4 representation).[10]

ARCHITECTURES

Natural language understanding encompasses numerous diverse tasks such as question answering,textual entailment,document classification and semantic similarity assessment.[12]

Pre-training a language model on unlabelled text has seen recent progress.The ability of a model to learn effectively from unlabelled data is significant in lightening the dependence of supervised learning in natural language processing(NLP).[12] Language models can be pre-trained on unlabelled text data which is easier to obtain than labelled data.[2]

Unsupervised pre-training predominately performs a regularisation role for supervised training.[1] Unsupervised pre-training is a special case of semi-supervised learning.
Unsupervised pre-training enables the architecture to be fine-tuned in a final training phase with respect to a supervised training criterion.[1] Pre-training a model is beneficial as it initialises the models network into a region of the parameter space where optimisation is easier for supervised learning.[1]The beneficial effects of unsupervised learning do not diminish as supervised training is peformed.[1]. The pre-training phase helps capture linguistic information. Transformer networks capture linguistic structure which have a longer range in comparison to previous models such as RNNS and CNNS.[12]

SAFE-GPT has not undergone supervised learning and has potential for being fine-tuning on specialised chemical spaces. This could enhance its utility in specialised tasks.[10]

Decoder-only models have advantages over Encoder-Decoder frameworks such as model size is reduced significantly. The attention matrix for the decoder-only model is a full rank matrix since it is a triangular matrix. The encoder-decoder attention matrix may not be full rank. [2] The defects of the decoder-only language model applied in a seq2seq task are partially caused by the Attention Degeneration Problem (ADP) in its attention component.[2]

ALIGNED VS. UNALIGNED

Pre-trained language models are able to predict a subsequent token at incredible scale. This allows the models to learn general-purpose representations. Various methods of aligning language models enable the learnt representations to be transferred to most language understanding or generation tasks. The most common methods are instruction tuning through supervised fine tuning (SFT) and preference tuning via reinforcement learning from human feedback (RLHF).[9, 19] Current alignment methods require significant compute power and specialised labelled data to achieve ChatGPT-level performance.[19] It is argued that most of the knowledge in large language models is learnt during the process of pre-training, and only a minimal amount of data for instruction tuning is needed to teach a model to produce high quality specialised output.[19]

Deterministic methods outperform stochastic methods on all tasks except open-ended text generation using unaligned models. [14] Unaligned models are more dependent on decoding methods in comparison to aligned models.[14] Among the most common stochastic methods, temperature sampling often performs the best, especially when using unaligned models, except for open-ended text generation. [14]

DECODERS

A decoding method defines how the generated token sequence is derived from a probability estimation.[14] The performance rankings of various models will change based on the decoding methods they make use of.[14] This tells one that decoders influence the effectiveness of a model's output generation. Decoding methods connect the next-token predictors with the text generators. Decoding methods significantly contribute towards transforming large language models into practical task solvers.[14] Decoding algorithms are most often classified as either search or sampling algorithms. Search methods objective is to attain accurate generation in goal driven tasks (e.g. summarisation), however they suffer from repetitive outputs in open-ended text generation. Sampling methods output more diverse text in open-ended text generation but suffer from unnatural topic drift.[8] Search methods can be seen as deterministic and sampling methods can be seen as stochastic.[5, 14]

Standard decoding methods, such as Beam and Greedy search are optimised for generating high likelihood sequence. [5] Beam and greedy search are classified as deterministic decoding methods. Stochastic decoding methods include Top-K and Temperature.[14] The choice between stochastic and deterministic decoders depends on the general task at hand. Specifically, if it is open-ended or directed generation. [5, 14]

Model alignment (fine-tuning the model) can reduce the size of effect between the different decoding methods. [14] Decoding methods effectiveness are also influenced by other factors such as the size of the model and quantisation. [14]

Deterministic methods produce fewer hallucinations and have a stronger ability to follow instructions compared to Stochastic methods.[14] Open-ended text generation tasks benefit from being able to generate a diverse set of candidate sequences. Which is the aim of stochastic methods.[5] Some tasks, such as goal-oriented dialogue, fall in between open-ended and directed generation.[4]

Some argue that decoding is not primarily at fault for neural text degeneration. Rather it is the Likelihood objective which is the root cause leading model assigning too much probability to sequences containing repeated and frequent.[17]

Restricting the generated output can involve constraining the decode. Lexical constraints objective is for the outputted text to satisfy the lexical constraints given. [3] It is often implemented through Beam Search. There are various implementation but they all involve the constraints criteria and how the beams are chosen and traversed with the criteria in mind.[3]

Structure and form constraints can also be implemented. These constraints are very complex and have more dimensions to consider in comparison to lexical constraints. Examples would be a poetry generator.[18]

DIRECTED SYSTEMS

Directed generation, or close-ended generation tasks are defined through (input, output) pairs. In directed generation the output is a constrained transformation of the input.[4]

Generally, closed-ended tasks favour deterministic methods. [14] Deterministic decoding methods perform better than stochastic decoding methods in directed generation tasks such as summerisation, machine translation and data-to-text generation.[4]

A frequent decoding objective, for directed generation, is maximisation-based decoding.[8] In directed generation the output is tightly scoped by the input. Repetition and genericness are not as problematic or concerning compared to open-ended generation.[4] Stochastic methods with self-consistency have been shown to outperform deterministic methods, albeit they require multiple runs.[14]

OPEN-ENDED SYSTEMS

In open-ended generation the input context restricts the space of acceptable output generations.The acceptable possible space in which an open-ended systems decoder can sample from is large. Thus there is large degree of freedom in what can plausibly come next, unlike in directed generation settings which begines with a tightly scoped search space.[4] Examples of open-ended generation, include conditional story generation and contextual text continuation. Thus, Open-ended generation favour using stochastic methods.[14] This is evident where beam search, a deterministic method, is ill-equipment in generating a list diverse candidate sequences- The candidates outputted from a large-scale beam search often only differ minor morphological variations or punctuation.[5] This shows that for open-ended text generation, maximisation is an inappropriate decoding objective.[4]

Moreover, decoding strategies such as beam search which optimise for output with high probability, , result in text that is heavily degenerate, even when implemented with powerful models such as GPT2 Large.[4]

COMPARING DIFFERENT DECODERS

Deterministic

Greedy. No current sub-exponential algorithm exists to find the optimal decoded sequence; therefore, approximations are used. The Arg-max/ greedy function selects the token with the highest probability at each timestep.[5, 14] The greedy algorithm produces short and repetitive output sequences. It does not permit multiple samples. It is seldom made use of in language modelling.[5]

$$x = \arg\max_{x_i} P(x_i|x_{<i}) \tag{1}$$

Beam search. Computing the overall most likely output sequence is intractable.[5] Beam search performs breadth-first search over a search space which is restricted this allows it to approximate finding the most likely sequence. At each decoding step, the algorithm maintains record of the W most probable hypotheses. The next set of partial hypotheses are chosen by expanding every path from the existing set of W hypotheses, and then choosing the W with the highest scores.[5, 14] Beam search only evaluates a subset of the overall search space. Beam search outputs multiple high-likelihood sequence which differ only by minor variants and thus the outputted sequences only differ in punctuation and minor morphological changes.[5]

Contrastive Decoding. Contrastive decoding is a search-based decoding method. Contrastive Decoding searches for text that maximises the difference between the log-probabilities of two models, the expert and amateur models. This method is subject to plausibility constraints which is where the search space is restricted to tokens which have sufficiently high probability under the expert LM.

Contrastive decoding requires zero additional training. The ability to contrast two language models of different sizes, allows for decoding higher quality text in comparison to a single larger LM. The amateur model is cheap to run and incurs very little inference time overhead.[8] Contrastive decoding avoids topic drift experienced by stochastic methods by using deterministic methods.[8]

$$\hat{x} = \underset{x \in \mathcal{X}}{\operatorname{argmax}}(\log\left(\text{Likelihood}_{\text{expert}}(x|x_{1:i-1})\right) - \log\left(\text{Likelihood}_{\text{amateur}}(x|x_{1:i-1})\right)) \quad (2)$$

Explanation:

(1) For each possible token x in the set $\mathcal{X}$:
   (a) Calculate the likelihood of token x according to the expert language model given the prefix $x_{1:i-1}$.
   (b) Calculate the likelihood of token x according to the amateur language model given the same prefix $x_{1:i-1}$.
   (c) Take the logarithm of the likelihoods obtained in steps (a) and (b).
   (d) Subtract the logarithm of the amateur likelihood from the logarithm of the expert likelihood.
(2) Select the token $\hat{x}$ that maximizes the difference obtained in step 1.

[8]

Stochastic

Temperature Sampling. Samples tokens from the estimated next-token distributions. The skewness of distributions can be controlled using a temperature hyperparameter τ.[13, 14]

$$P(x_i|x_{<i}) = \frac{e^{z_i/\tau}}{\sum_{j=1}^{V} e^{z_j/\tau}}$$

where:

- $P(x_i|x_{<i})$ is the probability of generating token x_i given the previous tokens $x_{<i}$.
- $e^{z_i/\tau}$ is the exponential function applied to the logit z_i divided by the temperature τ.
- $\sum_{j=1}^{V} e^{z_j/\tau}$ is the summation of the exponential functions over all tokens in the vocabulary.
- z_i is the logit (raw output score) corresponding to token x_i.
- τ is the temperature parameter.
- V is the size of the vocabulary, i.e., the number of possible tokens.

[13]

Nucleus Sampling. Sampling directly from the probabilities predicted by the model is impacted by the unreliable tails of the distribution- this results in text that is incoherent and possibly unrelated to the context. This led to Top-P/nucleus sampling where, the minimal set of most probable tokens that cover a specified percentage p of the distribution are considered during sampling.[14]

Given a distribution $P(x|x_{1:i-1})$, we define its top-p vocabulary $V^{(p)} \subseteq V$ as the smallest set such that

$$\sum_{x \in V^{(p)}} P(x|x_{1:i-1}) \geq p.$$

Let $P' = \sum_{x \in V^{(p)}} P(x|x_{1:i-1})$. The original distribution is re-scaled to a new distribution, from which the next word is sampled:

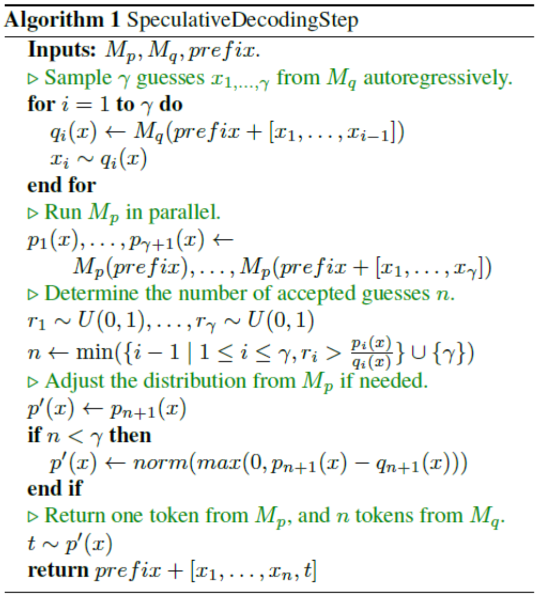
$$P'(x|x_{1:i-1}) = \begin{cases} \frac{P(x|x_{1:i-1})}{p'} & \text{if } x \in V^{(p)}, \\ 0 & \text{otherwise.} \end{cases}$$

[4]

Top-k Sampling. Samples from the top-k probable tokens. [14]

Similar to Nucleus.But differs where it truncates distribution from which it samples. The distribution is re-scaled as in equation for nucleus, and sampling is performed based on that distribution.[4]

Speculative Decoding. Speculative Decoding allows one to sample from auto-regressive models faster by computing tokens in parallel. This is done by speculative execution sequentially.[7] This is approached through a stochastic framework. This decoding uses its own sampling and decoding technique.[7]



Algorithm 1 SpeculativeDecodingStep

```
Inputs: $M_p, M_q, prefix$.
▷ Sample $\gamma$ guesses $x_{1,\ldots,\gamma}$ from $M_q$ autoregressively.
for $i = 1$ to $\gamma$ do
  $q_i(x) \leftarrow M_q(prefix + [x_1, \ldots, x_{i-1}])$
  $x_i \sim q_i(x)$
end for
▷ Run $M_p$ in parallel.
$p_1(x), \ldots, p_{\gamma+1}(x) \leftarrow$
    $M_p(prefix), \ldots, M_p(prefix + [x_1, \ldots, x_\gamma])$
▷ Determine the number of accepted guesses $n$.
$r_1 \sim U(0,1), \ldots, r_\gamma \sim U(0,1)$
$n \leftarrow \min(\{i - 1 \mid 1 \leq i \leq \gamma, r_i > \frac{p_i(x)}{q_i(x)}\} \cup \{\gamma\})$
▷ Adjust the distribution from $M_p$ if needed.
$p'(x) \leftarrow p_{n+1}(x)$
if $n < \gamma$ then
  $p'(x) \leftarrow norm(max(0, p_{n+1}(x) - q_{n+1}(x)))$
end if
▷ Return one token from $M_p$, and $n$ tokens from $M_q$.
$t \sim p'(x)$
return $prefix + [x_1, \ldots, x_n, t]$
```

[7]

Figure 3: Speculative Decoding Algorithm

Decoders speed

Stochastic decoding are able to achieve similar decoding speeds to greedy search. Beam search in addition to other advanced deterministic methods are considerably slower relative to greedy search. There is a discrepancy in speed becoming more prominent as the generation increases in length for some of those methods.[14]In general inference from auto-regressive models is slow, as it is often done sequentially.[15]

CONCLUSION

Throughout this literature review, different decoding strategies were analysed and various paradigms of when specific decoding techniques are effective were viewed.

It is evident that the problem of molecular generation must make use of a stochastic or deterministic decoding method. One would hypothesis that the requirements of molecular generation being diversity, uniqueness and validity would favour stochastic methods, as they are favoured in open-ended generation tasks, in comparison to the deterministic methods which are used in directed generation tasks. The large degree of freedom in possible output for open-ended generation tasks would be more appealing for a molecular generation task, as maximising diversity and uniqueness allow for a greater set of molecules to be produced. Selecting the best decoding method would improve the models effectiveness at producing the desired output.

REFERENCES

[1] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. 2010. Why does unsupervised pre-training help deep learning?. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 201–208.

[2] Zi-Yi Fu, Wai Lam, Qin Yu, A.M.C. So, Shiyang Hu, Zhiyuan Liu, and Nigel Collier. 2023. Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder. *arXiv preprint arXiv:2304.04052* (2023).

[3] Chris Hokamp and Qiang Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138* (2017).

[4] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751* (2019).

[5] David Ippolito, Rene Kriz, Masha Kustikova, João Sedoc, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. *arXiv preprint arXiv:1906.06362* (2019).

[6] Mario Krenn, Florian Häse, Ankur Nigam, Pascal Friederich, and Alán Aspuru-Guzik. 2020.

[7] Yossi Leviathan, Michal Kalman, and Yael Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*. PMLR, 19274–19286.

[8] Xiaolei Li, Ari Holtzman, David Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097* (2022).

[9] Benyou Lin, Abhijeet Ravichander, Xianchao Lu, Nouha Dziri, Mark Sclar, Kavya Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552* (2023).

[10] Elaheh Noutahi, Chiara Gabellini, Matthew Craig, Jia Siang Lim, and Pulchérie Tossou. 2023. Gotta be SAFE: A New Framework for Molecular Design. *arXiv preprint arXiv:2310.10773* (2023).

[11] Dmitry Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oleg Tatanov, Sergey Belyaev, Raul Kurbanov, Aleksandr Artamonov, Vladimir Aladinskiy, Mark Veselov, and Artur Kadurin. 2020. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology* 11 (2020), 565644.

[12] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[13] M. Renze and E. Guven. 2024. The Effect of Sampling Temperature on Problem Solving in Large Language Models. (2024). arXiv:2402.05201 [cs.CL]

[14] Chuan Shi, Han Yang, Deng Cai, Zhan Zhang, Yan Wang, Yiming Yang, and Wai Lam. 2024. A thorough examination of decoding methods in the era of llms. *arXiv preprint arXiv:2402.06925* (2024).

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, Vol. 30.

[16] David Weininger. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* 28, 1 (1988), 31–36.

[17] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319* (2019).

[18] Sebastian Zarrieß, Hannah Voigt, and Sebastian Schüz. 2021. Decoding methods in neural language generation: a survey. *Information* 12, 9 (2021), 355.

[19] Changhan Zhou, Peng Liu, Pengcheng Xu, Srinivasan Iyer, Jian Sun, Yuning Mao, Xuezhe Ma, Aviv Efrat, Peng Yu, Lillian Yu, and Shujie Zhang. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems* 36 (2024).