Estimating Text Temperature

Anonymous ACL submission

Abstract

Autoregressive language models typically use temperature parameter at inference to shape the probability distribution and control the randomness of the text generated. After the text was generated, this parameter can be estimated using maximum likelihood approach. Following it, we propose a procedure to estimate the temperature of any text, including ones written by humans, with respect to a given language model. We evaluate the temperature estimation capability of a wide selection of small-to-medium LLMs. We then use the best-performing Qwen3 14B to estimate temperatures of popular corpora.

1 Text Generation with a Language Model

Given an input text as a context, the goal of open-ended generation is to produce a coherent continuation of the text (Holtzman et al., 2020). More formally, given a sequence of m tokens $t^{(1)} \dots t^{(m)}$ as a context, the objective is to generate the next n continuation tokens, resulting in the completed sequence $t^{(1)} \dots t^{(m+n)}$. This is achieved through the use of the left-to-right text probability decomposition, which is used to generate the sequence one token at a time, using a particular decoding strategy.

A common approach to text generation is to shape a probability distribution through temperature (Ackley et al., 1985). Let

- $u^{(i)}$ be the vector of logits at the step i.
- $u^{(i)}_{obs}$ be the specific logit value corresponding to the token $t^{(i)}$.

The probability of observing the specific token at step i is then re-estimated as:

$$p(t^{(i)}|u^{(i)}, T) = \frac{\exp(u^{(i)}_{obs}/T)}{\sum_l \exp(u^{(i)}_l/T)} \quad (1)$$

Setting $T \in [0, 1)$ skews the distribution towards high-probability events, and, similarly, $T \in (1, \infty)$ skews the distribution towards low-probability events.

2 Estimating Temperatures

2.1 Estimating the Temperature of a Generated Sequence

Suppose that we have logits for multiple generation steps and actual generated tokens for the same steps. What is the maximum likelihood estimate of temperature T? We can treat it as a classic parameter estimation problem.

The total log-likelihood $\mathcal{L}(T)$ for all N steps is the sum of the individual log-probabilities:

$$\mathcal{L}(T) = \sum_{i=1}^{N} \log \left(\frac{\exp(u^{(i)}_{obs}/T)}{\sum_l \exp(u^{(i)}_l/T)} \right) \quad (2)$$

To find the MLE, we take the derivative of $\mathcal{L}(T)$ with respect to T and set it to zero.

$$\frac{d\mathcal{L}}{dT} = \sum_{i=1}^{N} \left[-\frac{u^{(i)}_{obs}}{T^2} - \left(-\frac{1}{T^2} \sum_l u^{(i)}_l p(l|u^{(i)}, T) \right) \right] = \frac{1}{T^2} \sum_{i=1}^{N} \left[\mathbb{E}[u^{(i)}|T] - u^{(i)}_{obs} \right] \quad (3)$$

Setting the derivative (3) to zero, we get for the MLE $\hat{T}$:

$$\sum_{i=1}^{N} u^{(i)}_{obs} = \sum_{i=1}^{N} \mathbb{E}[u^{(i)}|\hat{T}], \quad (4)$$

where

$$\mathbb{E}[u^{(i)}|\hat{T}] = \sum_l u^{(i)}_l \frac{\exp(u^{(i)}_l/\hat{T})}{\sum_k \exp(u^{(i)}_k/\hat{T})} \quad (5)$$

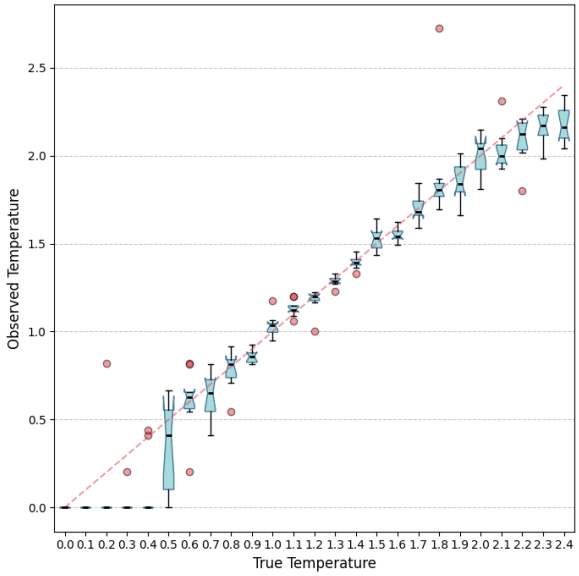

Figure 1: Estimated vs. generation temperature for granite-4.0-micro.

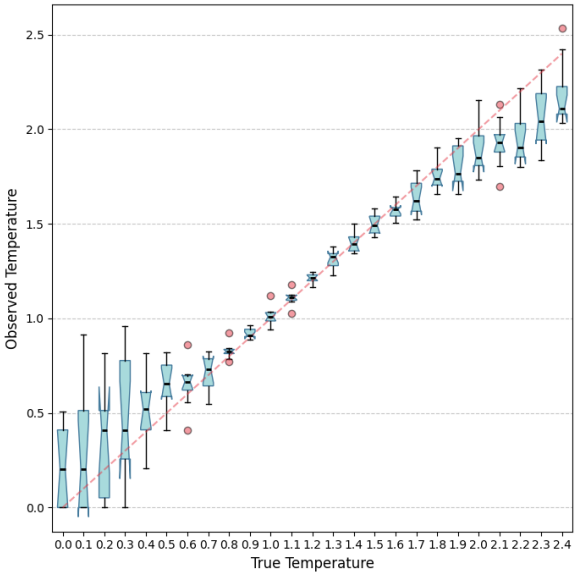

Figure 2: Estimated vs. generation temperature for Llama-3.2-3B.

Thus, the maximum likelihood estimate for T is the temperature at which the sum of the observed logits equals the sum of the expected logits predicted by the model.

2.2 Estimating the Temperature of Any Text

Now, observe that the text we work on in the previous subsection does not need to be a generated one. As soon as we have tokenized a text and have estimated the probability distributions for all the tokens in the text with some probabilistic model (for example, an LLM), we can estimate the text temperature using (4). This estimate will, of course, depend on the LLM model used.

To specifically find the temperature, we numerically solve equation (4) for T, given the logits for all the tokens and specific tokens present in the text.

3 Estimating Temperature of LLM-Generated Texts

To better understand the properties of the proposed temperature estimation approach, we generate texts with an assortment of LLMs in small to medium sizes representing different LLM families, finetuning levels and architectures:

- Qwen3 0.6B,1.7B 4B, 8B, 14B (Yang et al., 2025) (normal, base and 4B-Thinking-2507)
- Phi-4-mini (Microsoft Team, 2025) (reasoning and instruct)
- Llama 3.1 8B, 3.2 1B, 3B, 11B-Vision (LLama Team, 2024) (base and instruct)
- gemma-3 270m, 1b, 4b, 12b (Gemma Team, 2025)
- DeepSeek-R1 distills: 0528-Qwen3-8B, Qwen-1.5B, Qwen-7B, Qwen-14B, and Llama-8B (DeepSeek AI, 2025)
- granite-4.0-micro (IBM Research, 2025) (base and normal)

We use 4-bit BitsAndBytes NF4 quantized models through the HuggingFace Transformers library (Wolf et al., 2020). All the texts are generated from a single random seed-controlled token using batch inference. We generate 10 texts 200 tokens long for each temperature in a range from 0.001 to 2.401 with step 0.1. We do not use top-k, top-p or any other decoding parameters such as no-repeat. To estimate the temperature, we use SciPy (Virtanen et al., 2020) root finding function `root_scalar` for the reverse temperature with bracket $[10^{-2}, 10^{4}]$ and initial value $5 * 10^{3}$. To speed up the computations we compute softmax (1, 5) using PyTorch on GPU. The code was mostly written by Gemini.

3.1 What Does the Measured Temperature Say About the Generation Temperature?

In the first series of experiments, we estimate the temperature of texts generated by an LLM with the

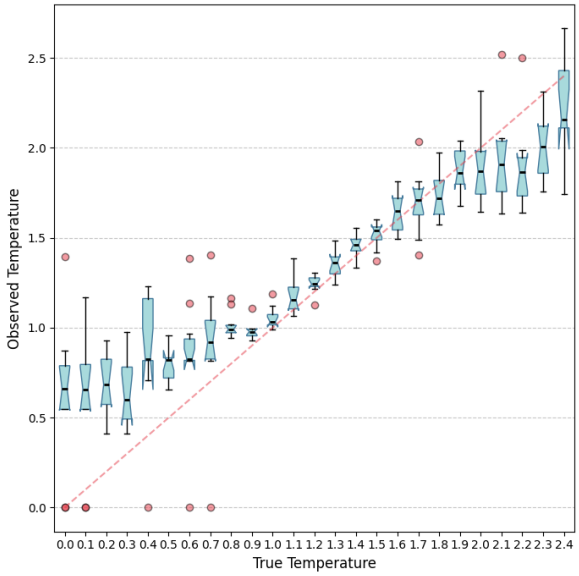

Figure 3: Qwen3-8B estimates Qwen3-4B

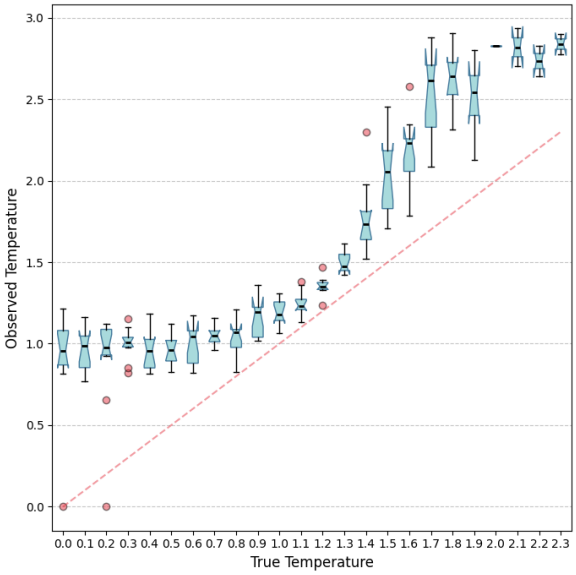

Figure 4: DeepSeek-R1-0528-Qwen3-8B estimates DeepSeek-R1-Distill-Qwen-1.5B

same LLM. For most LLMs, the estimated temperature insignificantly differs from the generation temperature.

For all the models, we observe pronounced and systematic differences between estimated and generation temperatures in areas of low and high temperatures (see Figures 1, 2). The difference in the low temperature area is the result of saturation when the generation with different temperatures produces one and the same result. Thus, the temperature estimation at low temperatures is an ill-posed problem that can be resolved using some kind of regularization yet to find. This constitutes an exciting area for future research. We don't yet have an explanation for the discrepancy between estimated and generation temperatures at high temperatures.

3.2 What Does the Measured Temperature say About the Model?

In the second series of experiments, we estimate the temperature with an LLM different from the one used to generate the text. The observed behaviors can be put into several qualitatively distinct groups. In one group, the graph of estimated vs. true temperature does not differ much from what a graph would be for the same LLM. This happens mostly for similar models in the families of Qwen, granite and LLama (see Figure 3 for an example).

The other group consistently overestimates the generation temperature (see Figure 4 for an example). This is typical for DeepSeek distills and happens in some other cases, but never for base models. We attribute this to narrower probability distributions of finetuned, especially reasoning, models.

Yet another group consists of model pairs that produce a pronounced S-shaped graph (see Figure 5). Finally, there are model pairs that show little to no correlation between the generation and the estimated temperatures (see Figure 6 for an example). This is often the case when either generator or estimator models are gemma of Phi.

To digitize these observations, we calculated statistical metrics of goodness such as MAE, R^2, Pearson p between observed and generation temperatures for each pair of models. Figure 7 presents a heatmap of MAE between the models. From this heatmap we can conclude that:

- Models in Qwen, LLama and granite families reasonably well estimate temperature of models from these families. They estimate even better when the generator and estimator models are from the same family.
- DeepSeek, gemma and Phi models are not good at both estimating and being estimated
- Base models are overall slightly better temperature estimators than finetuned ones. Base models better estimate base models and finetuned better estimate finetuned ones. This is especially pronounced for Llama family.
- Larger models are slightly better estimators

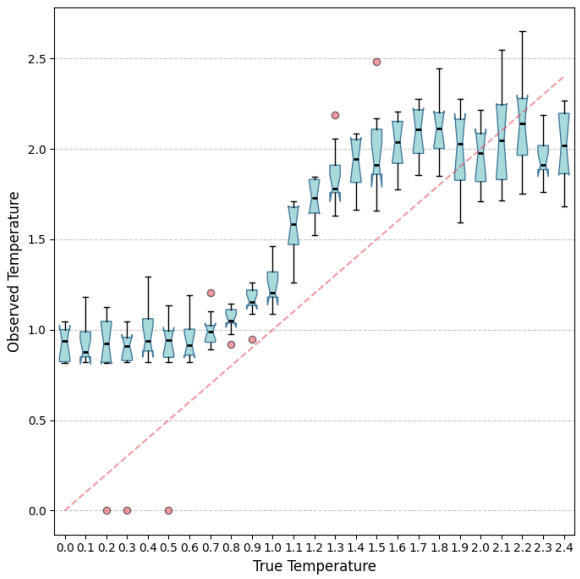

Figure 5: DeepSeek-R1-0528-Qwen3-8B estimates Meta-Llama-3.1-8B-Instruct

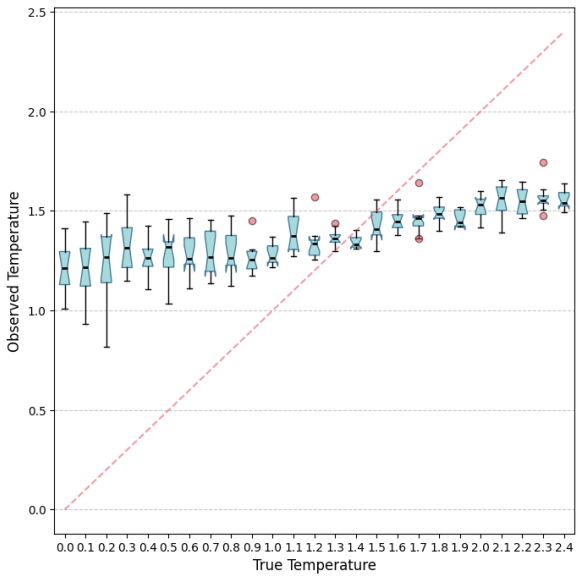

Figure 6: gemma-3-1b-it estimates Qwen3-14B

Dataset	Mean t	std
WikiText	1.0585	0.0645
Poetry	1.0089	0.1044
Jokes	1.1003	0.1037
GSM8K	1.0942	0.0992
Code/Python	0.9242	0.1539
IMDB	1.0244	0.0385
HH RLHF	1.0063	0.0641
AG News	1.1023	0.0734
Yelp	1.0349	0.0545

Table 1: Temperatures of corpora

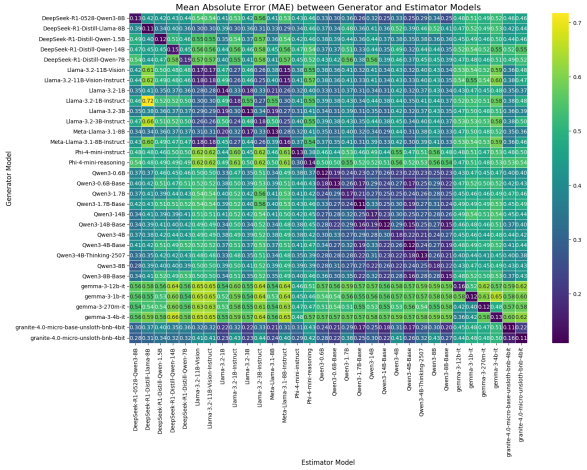

Figure 7: Heatmap of MAE between models

- Qwen models are the best estimators, the absolute leader being Qwen3-14B-base

4 Estimating Temperature of Human-written Corpora

Based on the results in the previous section, we use Qwen3 14B Base as the preferred LLM for the temperature estimation. We estimate temperature of several diverse datasets: WikiText-103 (Merity et al., 2016), Poetry dataset (Unknown, 2025b), Jokes dataset (Unknown, 2025a), GSM8K (Cobbe et al., 2021), Python code (Unknown, 2025c), IMDB (Maas et al., 2011), HH RLHF (Bai et al., 2022; Ganguli et al., 2022), AG News (Zhang et al., 2015b; Gulli, 2005; del Corso et al., 2005), Yelp (Zhang et al., 2015a). We sample 300 texts from each dataset and average temperatures over texts in a dataset. The results are in the Table 1.

The estimated temperatures are all close to 1. This is unsurprising. Notable exceptions are Jokes, GSM8K and AG News that have a statistically significant higher temperature of 1.1, and Python code that has a lower temperature of 0.9. Explaining these temperature phenomena is an exciting topic for future research.

5 Conclusion

We proposed a procedure to estimate the temperature of any text, with respect to a given language model. We evaluated the temperature estimation capability of a wide selection of small-to-medium LLMs. While the temperatures measured for most corpora are close to 1, the suggested method spots some interesting temperature phenomena.

Limitations

The experiments were conducted for English only. This may limit generalization to other languages.

The list of LLM tested is limited to small-to-medium models. Scaling laws, if any, are unclear.

References

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek DeepSeek AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Gianna Maria del Corso, Antonio Gullì, and Francesco Romani. 2005. Ranking a stream of news. In *The Web Conference*.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom Brown, Nicholas Joseph, Sam McCandlish, Chris Olah, Jared Kaplan, and Jack Clark. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *Preprint*, arXiv:2209.07858.

Google Gemma Team. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

A. Gulli. 2005. The anatomy of a news search engine. In *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, WWW '05, page 880–881, New York, NY, USA. Association for Computing Machinery.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

IBM Research. 2025. Granite 4.0 language models. https://github.com/ibm-granite/granite-4.0-language-models. Accessed: 2025-10-01.

Meta LLama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Microsoft Microsoft Team. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *Preprint*, arXiv:2503.01743.

Unknown. 2025a. Jokes dataset.

Unknown. 2025b. Poetry dataset.

Unknown. 2025c. python code dataset.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015a. Character-level convolutional networks for text classification. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *NIPS*.