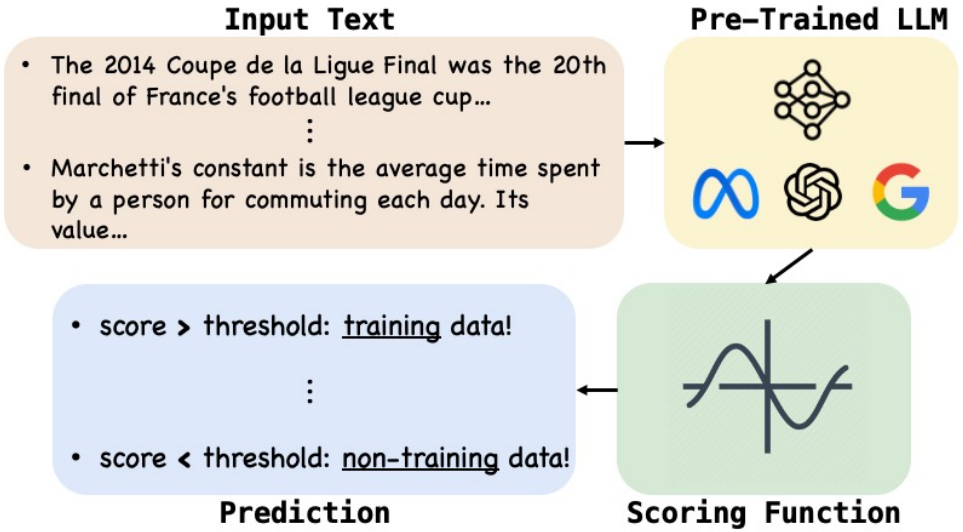
Min-K%++: Improved Baseline for Detecting Pre-Training Data from Large Language Models

Jingyang Zhang[1*], Jingwei Sun[1*], Eric Yeats[1], Yang Ouyang[1],
Martin Kuo[1], Jianyi Zhang[1], Hao Yang[1,2], Hai Li[1]
[1]Duke University, [2]Johns Hopkins University

Pre-training data detection for LLMs



Given a data instance x and the (gray-box) access to a pre-trained model $\mathcal{M}$, we want to give binary prediction of whether the sample x appears in the training set $\mathcal{D}$, with a scoring function s:

$$\text{prediction}(x, \mathcal{M}) = \begin{cases} 1\ (x \in \mathcal{D}), & s(x; \mathcal{M}) \geq \lambda \\ 0\ (x \notin \mathcal{D}), & s(x; \mathcal{M}) < \lambda \end{cases},$$

- This is an instance of Membership Inference Attack (MIA) that explicitly focuses on pre-trained LLMs, which poses unique challenges due to LLM pre-training setup.

- This problem has implications for critical issues such as copyright violation, test data contamination, and privacy auditing.

Revisiting the training objective

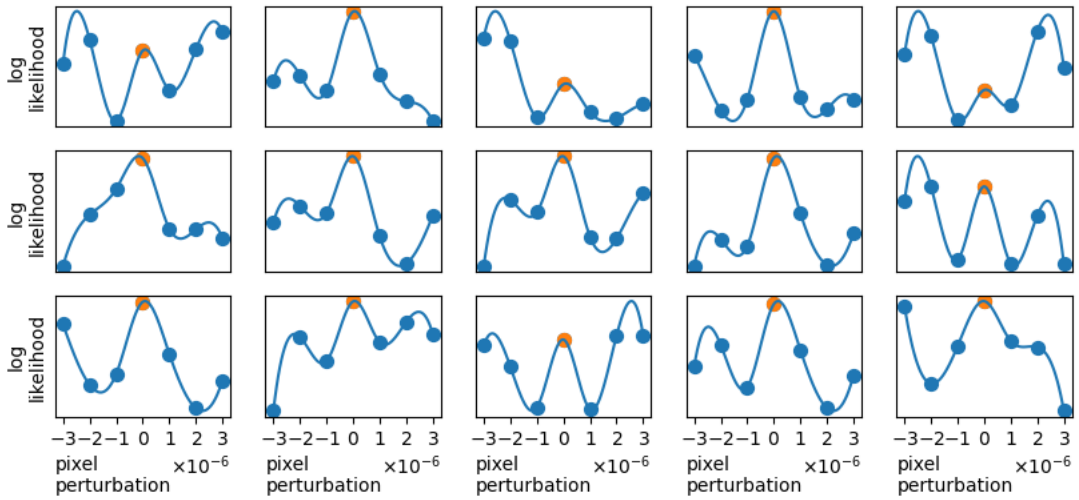
- LLM does maximum likelihood training, which is equivalent to *implicit score matching* (ISM).

- The minimization of ISM indicates that **training samples tend to form local maxima along each input dimension** in the likelihood space.

- This insight can be empirically confirmed by examining a pre-trained diffusion model.

$$\frac{1}{N}\sum_{\boldsymbol{x}}\left[\frac{1}{2}||\psi(\boldsymbol{x})||^2+\underbrace{\sum_{i=1}^{d}\frac{\partial\psi_i(\boldsymbol{x})}{\partial\boldsymbol{x}_i}}_{\text{the sum of the second-order partial derivatives}}\right],$$

$$\psi(\boldsymbol{x})=\frac{\partial\log p(\boldsymbol{x})}{\partial\boldsymbol{x}}$$



Translating the insight to a practical method

$$\text{Min-K\%++}_{\text{token seq.}}(x_{<t}, x_t) = \frac{\log p(x_t|x_{<t}) - \mu_{\cdot|x_{<t}}}{\sigma_{\cdot|x_{<t}}},$$

$$\text{Min-K\%++}(x) = \frac{1}{|\text{min-}k\%|} \sum_{(x_{<t},x_t)\in\text{min-}k\%} \text{Min-K\%++}_{\text{token seq.}}(x_{<t}, x_t).$$

$$\mu_{x_{<t}} = \mathbb{E}_{z\sim p(\cdot|x_{<t})}[\log p(z|x_{<t})] \qquad \sigma_{x_{<t}} = \sqrt{\mathbb{E}_{z\sim p(\cdot|x_{<t})}[(\log p(z|x_{<t}) - \mu_{x_{<t}})^2]}$$

- Analogous to detecting local maximum in continuous distribution, we propose to identify whether each token of the input sequence has larger probability than other candidates.

- Min-K%++ introduces no computation overhead and requires no extra model for detection.

Results

Improved performance on two benchmarks

Table 1: AUROC results on WikiMIA benchmark (Shi et al., 2024). *Ori.* and *Para.* denote the original and paraphrased settings, respectively. **Bolded** number shows the best result within each column across all methods. The proposed Min-K%++ leads to remarkable improvements over existing methods in most settings.

		Mamba-1.4B		Pythia-6.9B		LLaMA-13B		LLaMA-30B		LLaMA-65B		Average	
Len.	**Method**	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*
32	Loss	61.0	61.4	63.8	64.1	67.5	68.0	69.4	70.2	70.7	71.8	66.5	67.1
	Ref	62.2	62.3	63.6	63.5	57.9	56.2	63.5	62.4	68.8	68.2	63.2	62.5
	Lowercase	60.9	60.6	62.2	61.7	64.0	63.2	64.1	61.2	66.5	64.8	63.5	62.3
	Zlib	61.9	62.3	64.3	64.2	67.8	68.3	69.8	70.4	71.1	72.0	67.0	67.4
	Neighbor	64.1	63.6	65.8	65.5	65.8	65.0	67.6	66.3	69.6	68.7	66.6	65.8
	Min-K%	63.2	62.9	66.3	65.2	68.0	68.4	70.1	70.7	71.3	72.2	67.8	67.9
	Min-K%++	**66.8**	**66.1**	**70.3**	**68.0**	**84.8**	**82.7**	**84.3**	**81.2**	**85.1**	**81.4**	**78.3**	**75.9**
64	Loss	58.2	56.4	60.7	59.3	63.6	63.1	66.2	65.5	67.9	67.7	63.3	62.4
	Ref	60.6	59.6	62.4	62.9	63.4	60.9	69.0	65.4	73.4	71.0	65.8	63.9
	Lowercase	57.0	57.0	58.2	57.7	62.0	61.0	62.1	59.8	64.5	61.9	60.8	59.5
	Zlib	60.4	59.1	62.6	61.6	65.3	65.3	67.5	67.4	69.1	69.3	65.0	64.5
	Neighbor	60.6	60.6	63.2	63.1	64.1	64.7	67.1	66.7	69.6	69.5	64.9	64.9
	Min-K%	62.2	58.0	65.0	61.1	66.0	64.0	68.5	65.7	69.8	67.9	66.3	63.3
	Min-K%++	**67.2**	**63.3**	**71.6**	**64.8**	**85.7**	**78.8**	**84.7**	**74.9**	**83.8**	**74.0**	**78.6**	**71.2**
128	Loss	63.3	62.7	65.1	64.7	67.8	67.2	70.3	69.2	70.7	70.2	67.4	66.8
	Ref	62.0	61.1	63.3	62.9	62.6	59.7	71.9	70.0	73.7	72.0	66.7	65.1
	Lowercase	58.5	57.7	60.5	60.0	60.6	56.4	59.1	55.4	63.3	60.1	60.4	57.9
	Zlib	65.6	65.3	67.6	**67.4**	69.7	69.6	71.8	71.5	72.1	**72.1**	69.4	69.2
	Neighbor	64.8	62.6	67.5	64.3	68.3	64.0	72.2	67.2	73.7	70.3	69.3	65.7
	Min-K%	66.8	64.5	69.5	67.0	71.5	68.7	73.9	70.2	73.6	70.8	71.0	68.2
	Min-K%++	**68.8**	**65.6**	**70.7**	66.8	**83.9**	**76.2**	**82.6**	**73.8**	**80.0**	70.7	**77.2**	**70.6**

Table 2: AUROC results on the challenging MIMIR benchmark (Duan et al., 2024) with a suite of Pythia models (Biderman et al., 2023). In each column, the best result across all methods is **bolded**, with the runner-up underlined. †Ref relies on an extra reference LLM. ‡Neighbor induces significant extra computational cost than others (25× in this case), for which reason we don't run on the 12B model. Despite not requiring a reference model like the Ref method does, our Min-K%++ consistently achieves superior or comparable performance.

	Wikipedia					Github					Pile CC					PubMed Central				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	50.2	51.3	51.8	52.8	53.5	65.7	69.8	71.3	73.0	74.0	49.6	50.0	50.1	50.7	51.1	49.9	49.8	49.9	50.6	51.3
†Ref	**51.2**	**55.2**	**58.1**	**61.8**	**63.9**	63.9	67.1	65.3	64.4	63.0	49.2	**52.2**	**53.7**	**54.9**	**56.7**	**51.3**	**53.1**	**53.7**	**55.9**	**58.2**
Zlib	51.1	52.0	52.4	53.5	54.3	**67.4**	**71.0**	**72.3**	**73.9**	**74.8**	49.6	50.1	50.3	50.8	51.1	49.9	50.0	50.1	50.6	51.2
‡Neighbor	50.7	51.7	52.2	53.2	/	65.3	69.4	70.5	72.1	/	49.6	50.0	50.1	50.8	/	47.9	49.1	49.7	50.1	/
Min-K%	50.2	51.3	51.8	53.6	54.4	65.7	69.9	71.4	73.2	74.3	50.3	51.0	50.8	51.5	51.7	50.6	50.3	50.5	51.2	52.3
Min-K%++	49.7	53.7	55.1	58.0	61.1	64.8	69.6	70.9	72.8	74.2	**50.6**	51.0	51.0	53.0	53.5	50.6	51.4	52.4	54.2	55.4

	ArXiv					DM Mathematics					HackerNews					Average				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	51.0	51.5	51.9	52.9	53.4	48.8	48.5	48.4	48.5	48.5	49.4	50.5	51.3	52.1	52.8	52.1	53.1	53.5	54.4	54.9
†Ref	49.4	51.5	53.1	**55.8**	57.5	**51.1**	**51.1**	50.5	51.1	50.9	49.1	**52.2**	**55.1**	**57.9**	**60.6**	52.2	**54.6**	**55.6**	**57.4**	58.7
Zlib	50.1	50.9	51.3	52.2	52.7	48.1	48.2	48.0	48.1	48.1	49.7	50.3	50.8	51.2	51.7	52.3	53.2	53.6	54.3	54.8
‡Neighbor	50.7	51.4	51.8	52.2	/	49.0	47.0	46.8	46.6	/	50.9	51.7	51.5	51.9	/	52.0	52.9	53.2	53.8	/
Min-K%	**51.0**	**51.7**	52.5	53.6	54.6	49.4	49.7	49.5	49.6	49.7	**50.9**	51.3	52.6	53.6	54.6	**52.6**	53.6	54.2	55.2	55.9
Min-K%++	50.1	51.1	**53.7**	55.2	**58.0**	50.5	50.9	**51.7**	**51.6**	**51.9**	50.7	51.3	52.6	54.5	56.5	52.4	54.1	55.3	57.0	**58.7**

- Min-K%++ consistently exhibits improved detection rates over existing reference-free methods.
- Compared with reference-based methods, Min-K%++ is comparable and superior.

Conclusions

- We provide a theoretical insight that translates the training data detection problem into the identification of local maxima.
- We propose Min-K%++, a novel and well-performing method for LLM pre-training data detection.
- We demonstrate improved detection rates on two established benchmarks, setting a new baseline for the field.

Code and more information:
https://github.com/zjysteven/mink-plus-plus