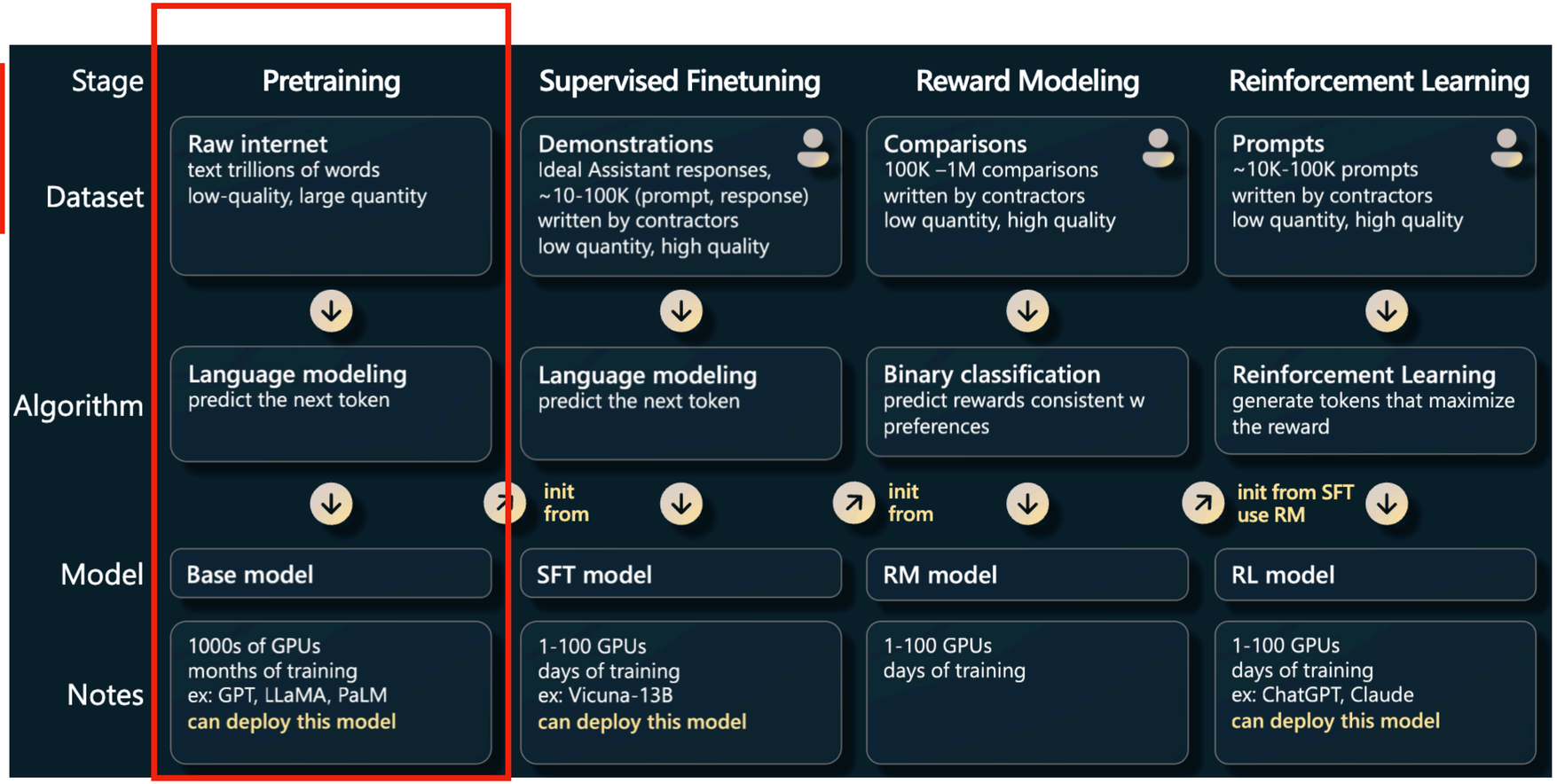
Automatic and Scalable Data Selection

Mining High-Quality Data to Speed up Language Model Pertaining

ZHANG Jipeng, 2023.11.18

LLM Evolving

- [Pre-Training] (Unsupervised Learning)
- [Instruction Tuning] (Supervised Fine-tuning)
- [Alignment] (RLHF)



Introduction: Why Data Selection

Introduction

The Importance of Quality Data in Large Language Models (LLMs)

Educational values deemed by the filter

High educational value

```python
import torch
import torch.nn.functional as F

def normalize(x, axis=-1):
    """Performs L2-Norm."""
    num = x
    denom = torch.norm(x, 2, axis, keepdim=True)
    .expand_as(x) + 1e-12
    return num / denom

def euclidean_dist(x, y):
    """Computes Euclidean distance."""
    m, n = x.size(0), y.size(0)
    xx = torch.pow(x, 2).sum(1, keepdim=True).
    expand(m, n)
    yy = torch.pow(x, 2).sum(1, keepdim=True).
    expand(m, m).t()
    dist = xx + yy - 2 * torch.matmul(x, y.t())
    dist = dist.clamp(min=1e-12).sqrt()
    return dist

def cosine_dist(x, y):
    """Computes Cosine Distance."""
    x = F.normalize(x, dim=1)
    y = F.normalize(y, dim=1)
    dist = 2 - 2 * torch.mm(x, y.t())
    return dist
```

Low educational value

```python
import re
import typing
...

class Default(object):
    def __init__(self, vim: Nvim) -> None:
        self._vim = vim
        self._denite: typing.Optional[SyncParent]
    = None
        self._selected_candidates: typing.List[int
    ] = []
        self._candidates: Candidates = []
        self._cursor = 0
        self._entire_len = 0
        self._result: typing.List[typing.Any] = []
        self._context: UserContext = {}
        self._bufnr = -1
        self._winid = -1
        self._winrestcmd = ''
        self._initialized = False
        self._winheight = 0
        self._winwidth = 0
        self._winminheight = -1
        self._is_multi = False
        self._is_async = False
        self._matched_pattern = ''
        ...
```

Introduction

The Importance of Quality Data in LLMs

Date	Model	Model size (Parameters)	Dataset size (Tokens)	HumanEval (Pass@1)	MBPP (Pass@1)
2021 Jul	Codex-300M [CTJ+21]	300M	100B	13.2%	-
2021 Jul	Codex-12B [CTJ+21]	12B	100B	28.8%	-
2022 Mar	CodeGen-Mono-350M [NPH+23]	350M	577B	12.8%	-
2022 Mar	CodeGen-Mono-16.1B [NPH+23]	16.1B	577B	29.3%	35.3%
2022 Apr	PaLM-Coder [CND+22]	540B	780B	35.9%	47.0%
2022 Sep	CodeGeeX [ZXZ+23]	13B	850B	22.9%	24.4%
2022 Nov	GPT-3.5 [Ope23]	175B	N.A.	47%	-
2022 Dec	SantaCoder [ALK+23]	1.1B	236B	14.0%	35.0%
2023 Mar	GPT-4 [Ope23]	N.A.	N.A.	67%	-
2023 Apr	Replit [Rep23]	2.7B	525B	21.9%	-
2023 Apr	Replit-Finetuned [Rep23]	2.7B	525B	30.5%	-
2023 May	CodeGen2-1B [NHX+23]	1B	N.A.	10.3%	-
2023 May	CodeGen2-7B [NHX+23]	7B	N.A.	19.1%	-
2023 May	StarCoder [LAZ+23]	15.5B	1T	33.6%	52.7%
2023 May	StarCoder-Prompted [LAZ+23]	15.5B	1T	40.8%	49.5%
2023 May	PaLM 2-S [ADF+23]	N.A.	N.A.	37.6%	50.0%
2023 May	CodeT5+ [WLG+23]	2B	52B	24.2%	-
2023 May	CodeT5+ [WLG+23]	16B	52B	30.9%	-
2023 May	InstructCodeT5+ [WLG+23]	16B	52B	35.0%	-
2023 Jun	WizardCoder [LXZ+23]	16B	1T	57.3%	51.8%
2023 Jun	**phi-1**	1.3B	7B	50.6%	55.5%

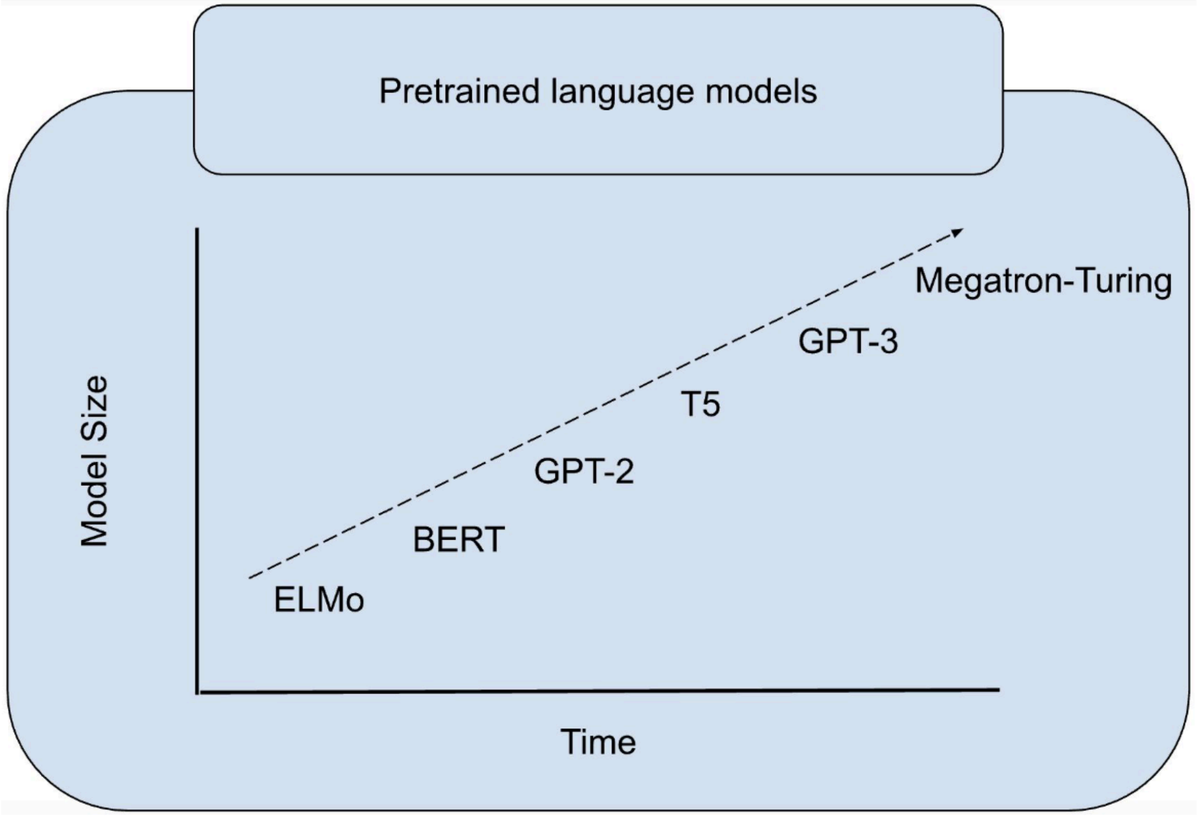
Smaller Model with **high quality data** can achieve better performance.

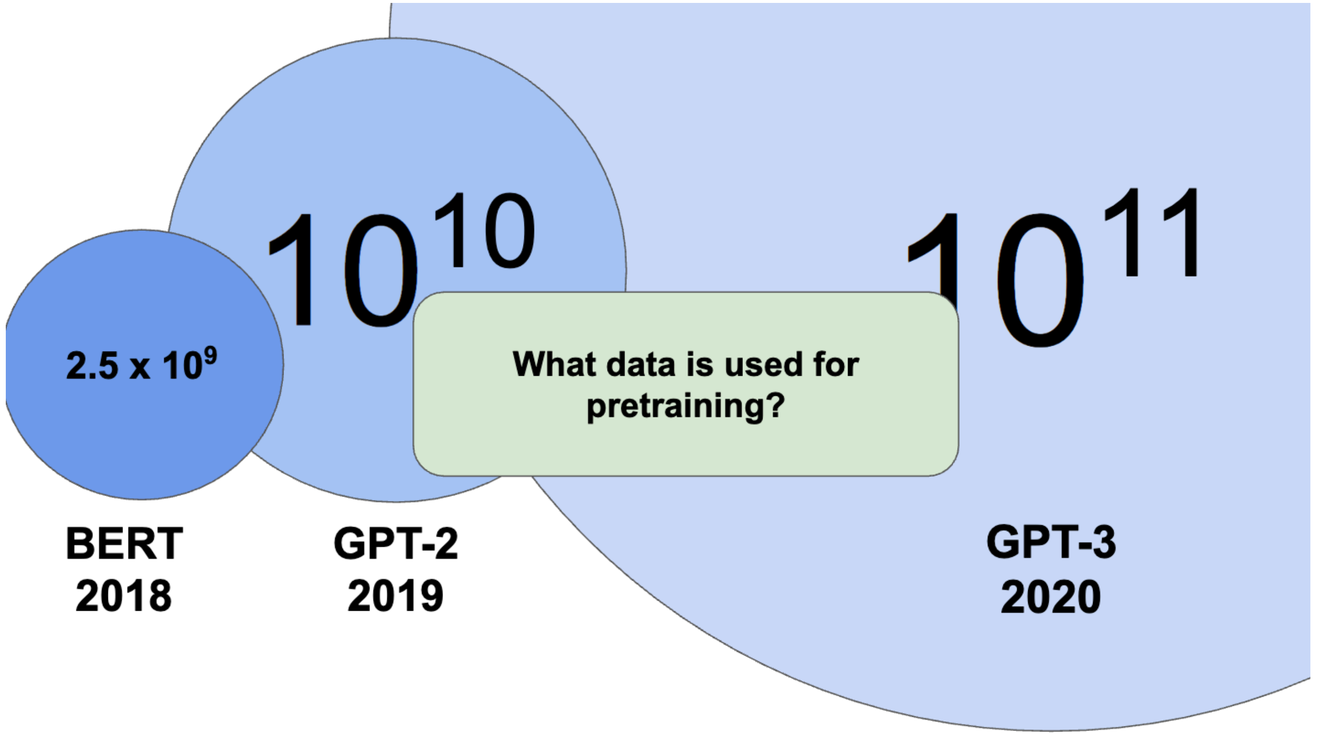
Both **quantity and quality** are important for language model training.

Introduction

The Challenges in Data Selection of Quality Data in LLMs

- There are many challenges in handling biased and noisy data, **computation scale** is the most significant one in LLM Training. We need **scalable** method.

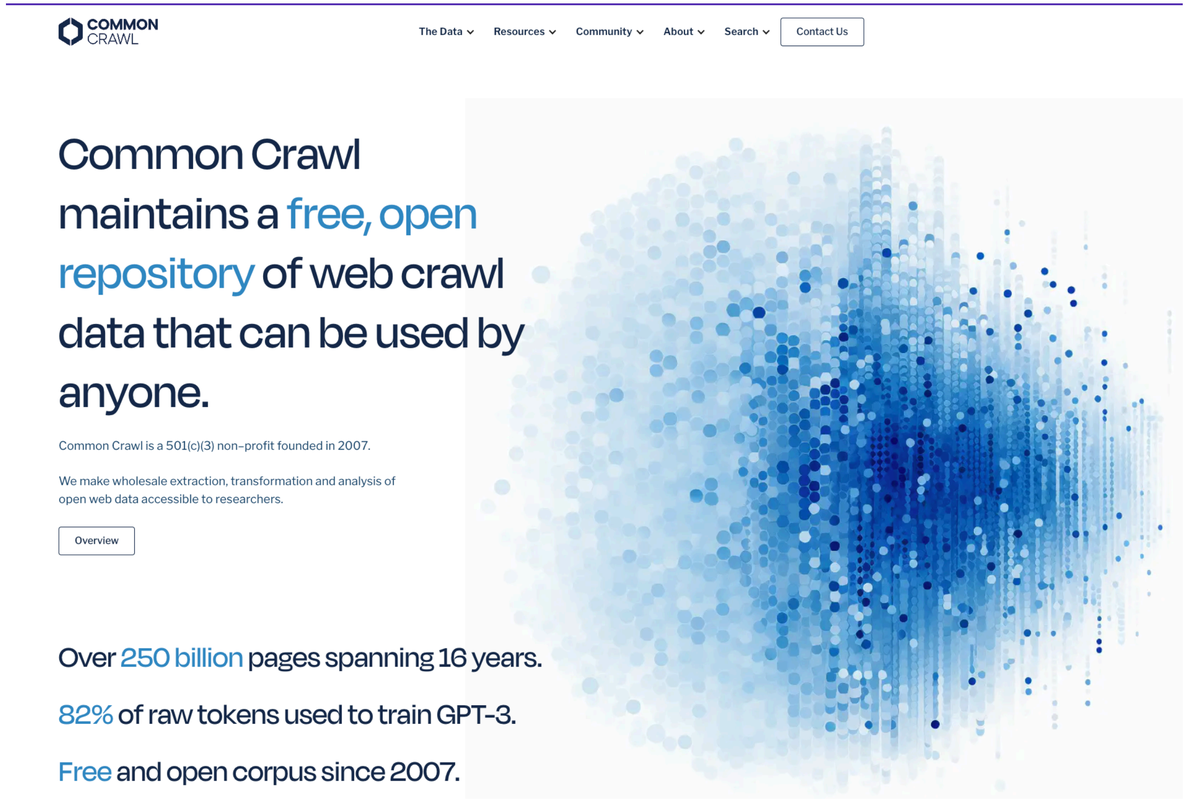




Introduction

The Challenges in Data Selection of Quality Data in LLMs

- Another potential challenge is the various sources of training data. We need **automatic** instead of rule-based heuristics to select data.

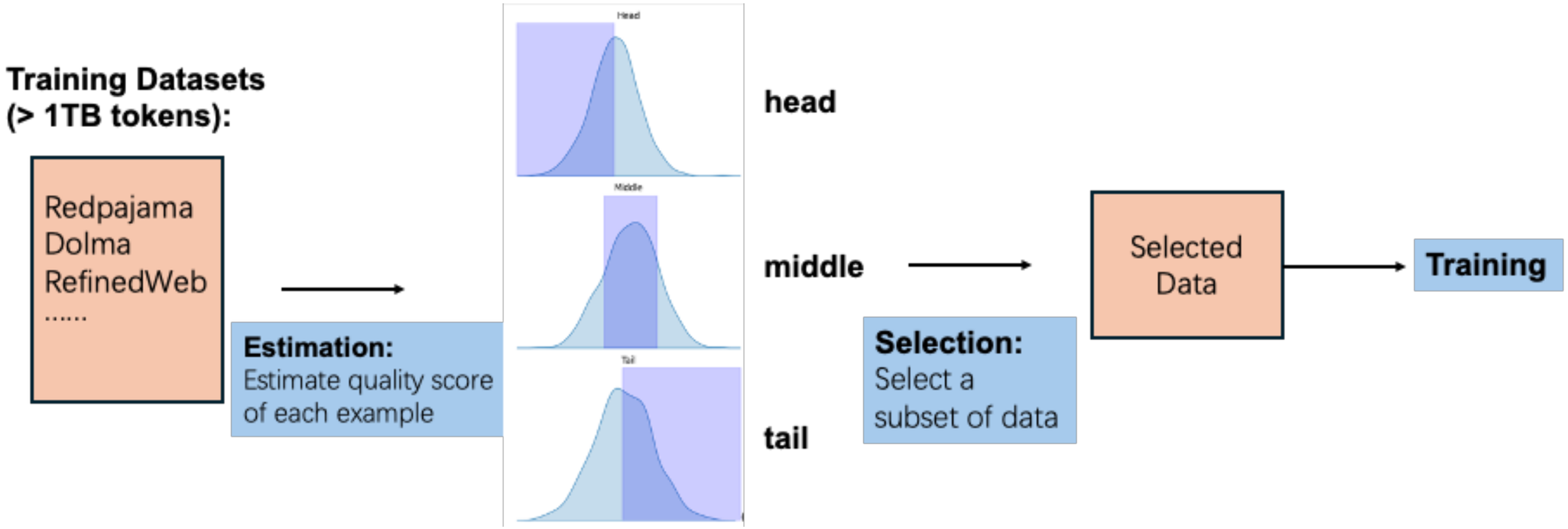


Newer datasets are typically built from Common Crawl, a **periodic internet-wide snapshot .**

Method: Measuring Data Quality

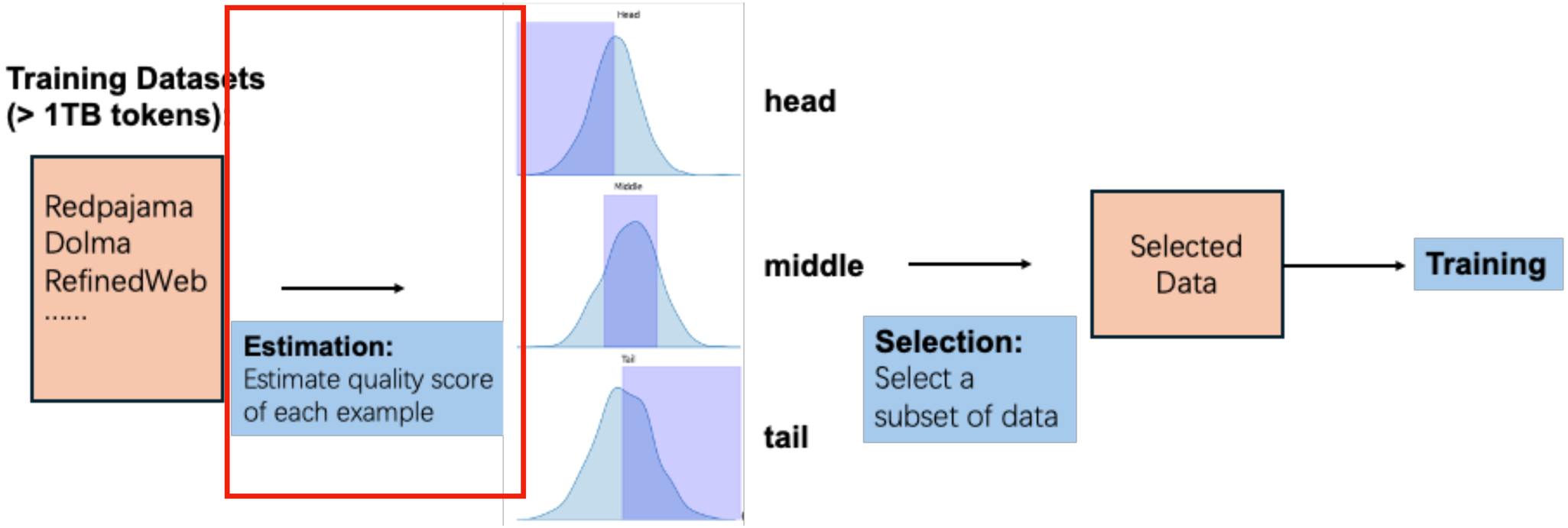
Method

Perplexity-Based Data Quality Estimation



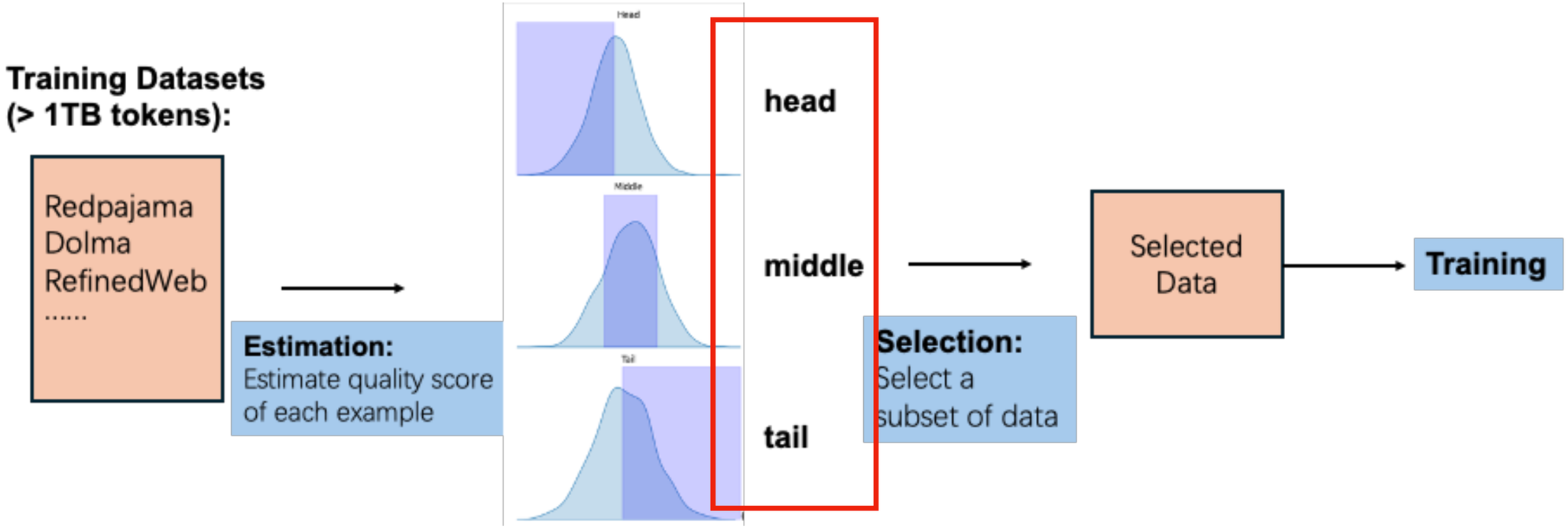
Method

Perplexity-Based Data Quality Estimation



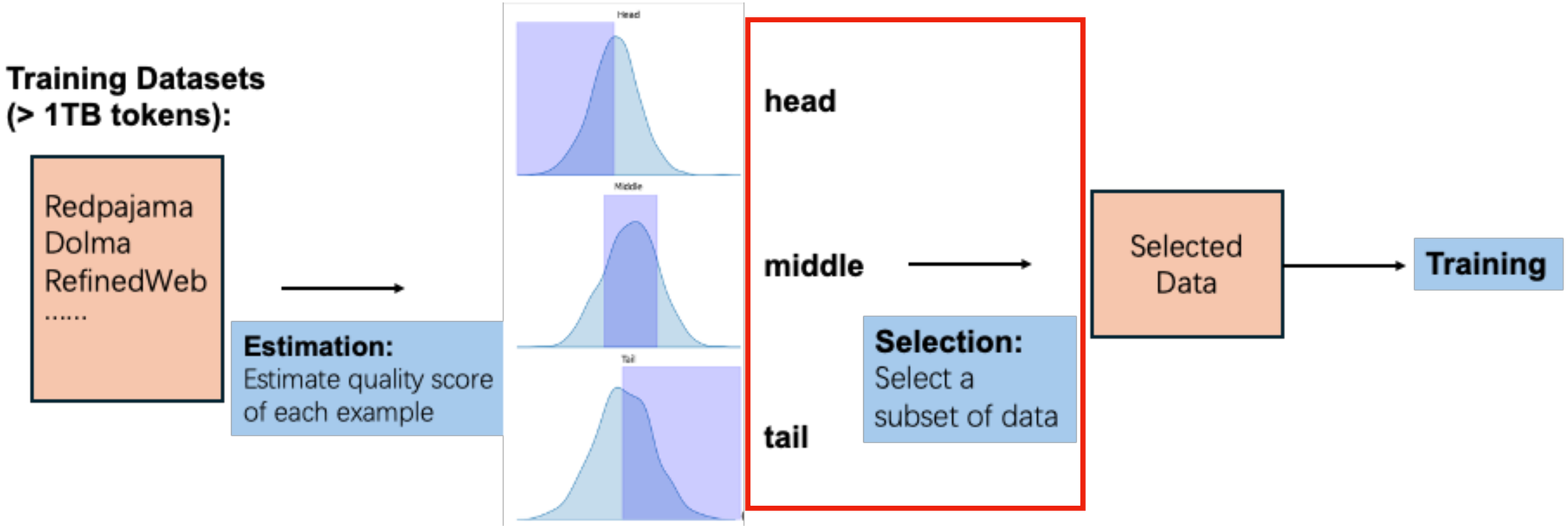
Method

Perplexity-Based Data Quality Estimation



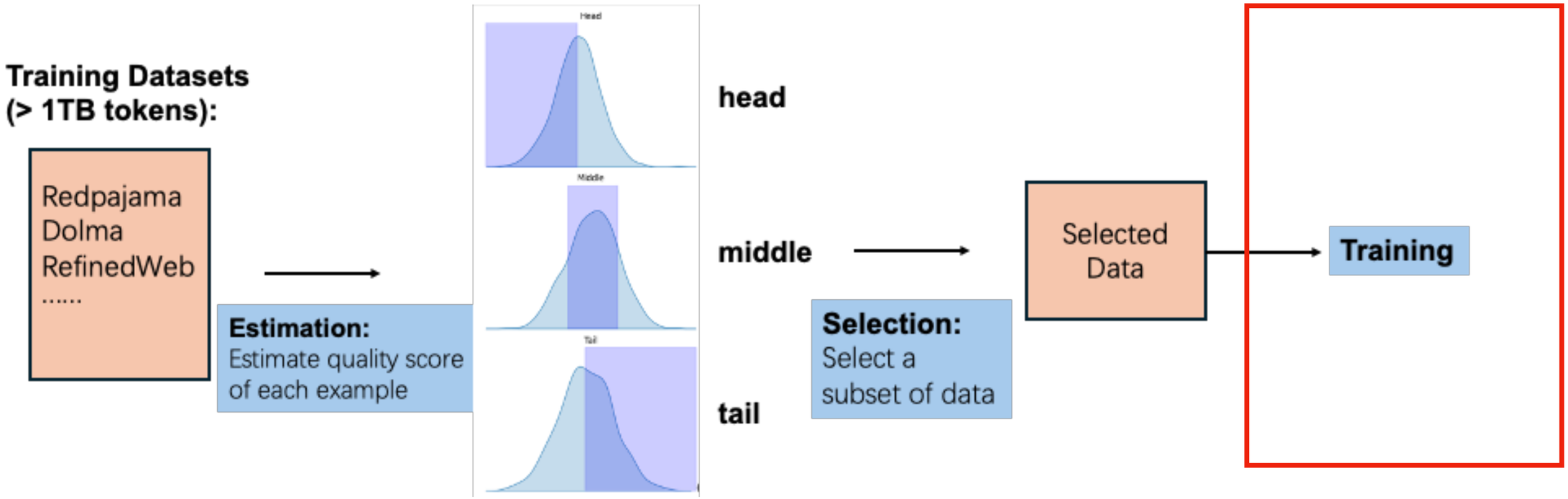
Method

Perplexity-Based Data Quality Estimation



Method

Perplexity-Based Data Quality Estimation



Method

Perplexity-Based Data Quality Estimation

- Negative Log Likelihood (NLL) is used for training LLM.

$$NLL = -\frac{1}{N}\sum_{i=1}^{N} log(p(sentence_i | context_i))$$

$$= -\frac{1}{N}\sum_{i=1}^{N} log(p(token_{i,1}, token_{i,2}, \ldots token_{i,n_i} | context_i))$$

-log **Pr**(“that is the question” | “To be, or not to be”) = 0

→ **Pr**(“that is the question” | “To be, or not to be”) = 1

→ **This model knows Shakespeare's Plays!**

Method

Perplexity-Based Data Quality Estimation

- Negative Log Likelihood (NLL) is used for training LLM.

$$NLL = -\frac{1}{N}\sum_{i=1}^{N} log(p(sentence_i | context_i))$$

$$= -\frac{1}{N}\sum_{i=1}^{N} log(p(token_{i,1}, token_{i,2}, \ldots token_{i,n_i} | context_i))$$

-log **Pr**("that is the question" | "To be, or not to be") = 0

→ **Pr**("that is the question" | "To be, or not to be") = 1

→ **This model knows Shakespeare's Plays!**

- A commonly used evaluation metric in NLP is Perplexity (PPL).

$$PPL = \frac{1}{N}\sum_{i=1}^{N} exp(-\frac{1}{n_i} \cdot log(p(sentence_i)))$$

These probabilties comes from models like
GPT-2-XL, GPT-Neo, Llama-7b, Llama-13b

Method

Observing Model Behavior to Select Data

- Confidence as the mean of model perplexity across steps' checkpoints

$$\mathrm{CONF}(M) = \frac{1}{|E|} \sum_{e=1}^{E} \mathrm{PPL}(M_e)$$

Method: Training Dynamics

Observing Model Behavior to Select Data

- Confidence as the mean of model perplexity across steps' checkpoints

$$\mathrm{CONF}(M) = \frac{1}{|E|}\sum_{e=1}^{E}\mathrm{PPL}(M_e)$$

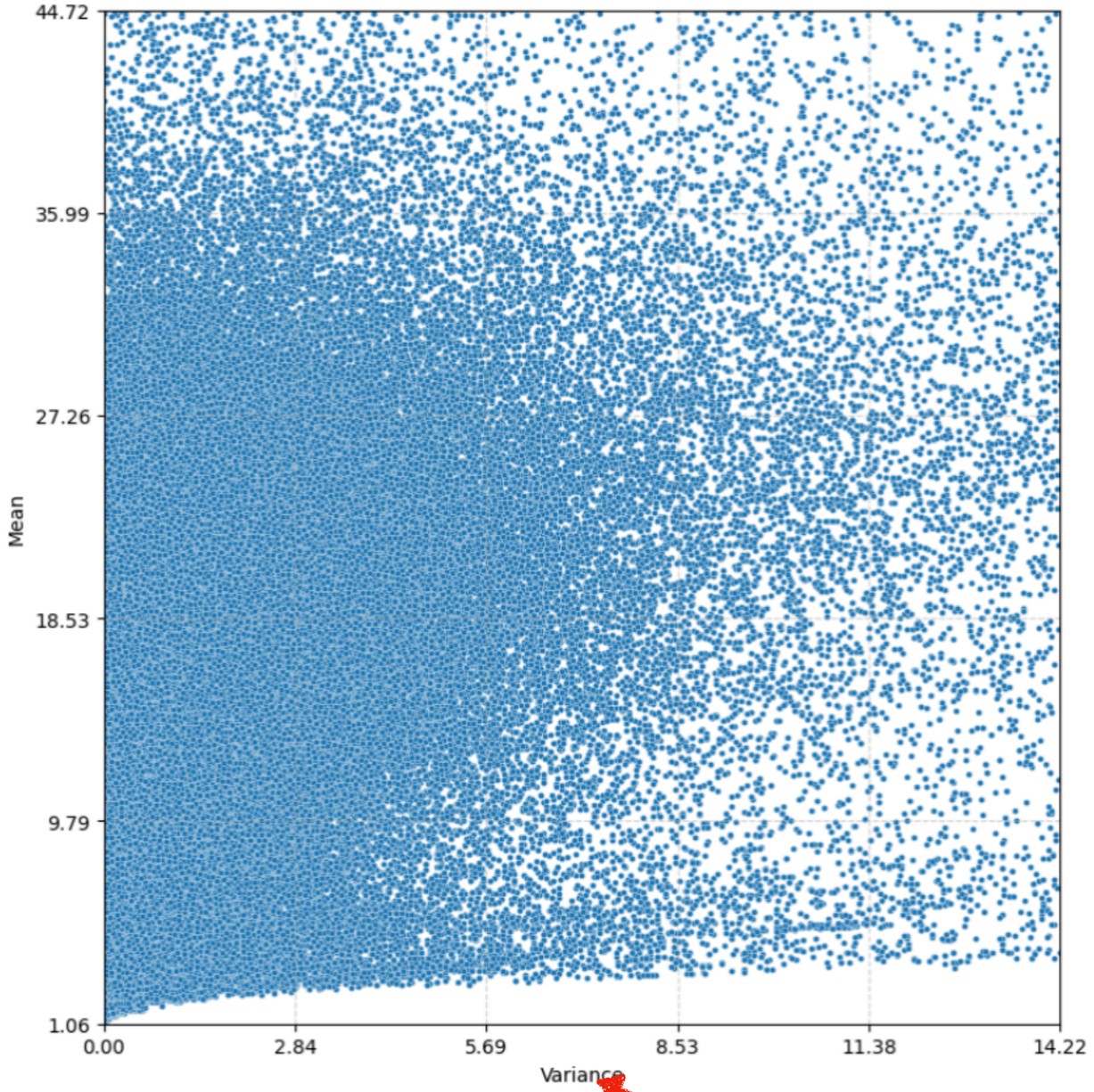
- Variability as the standard deviation of model perplexity across checkpoints

$$\mathrm{VAR}(M) = \sqrt{\frac{\sum_{e=1}^{E}(PPL(x) - \mathrm{CONF}(M_e))^2}{|E|}}$$

- With a model called Baichuan2-7b, we can access its intermediate checkpoints to compute its training dynamics.

Method

$$\mathrm{CONF}(M) = \frac{1}{|E|}\sum_{e=1}^{E}\mathrm{PPL}(M_e)$$



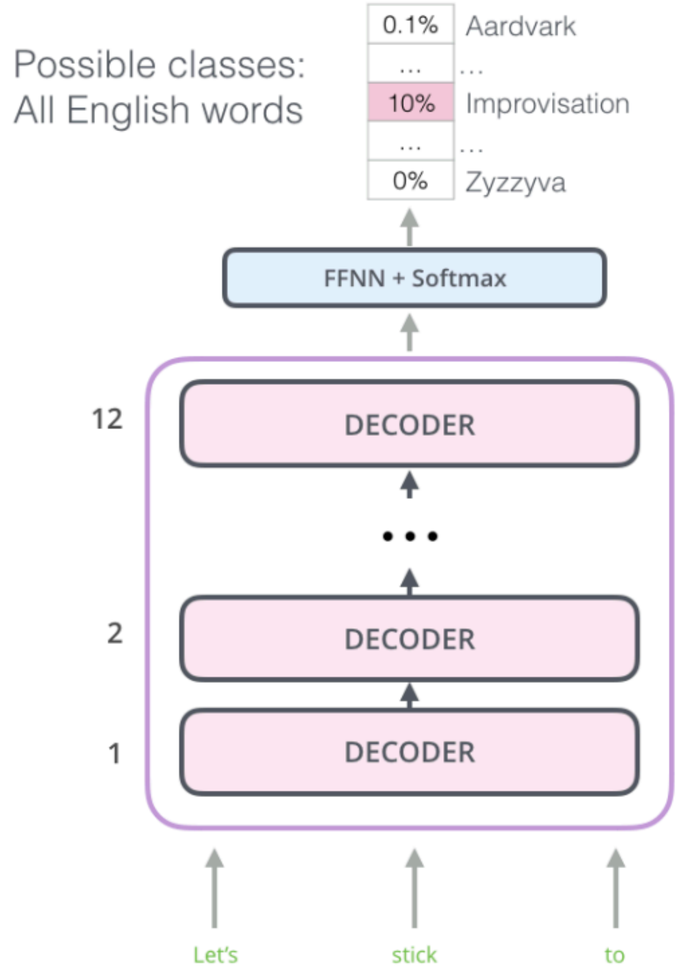
$$\mathrm{VAR}(M) = \sqrt{\frac{\sum_{e=1}^{E}(PPL(x) - \mathrm{CONF}(M_e))^2}{|E|}}$$

Experiment: Implementation

Experiment

Proxy Training Model GPT2

- Model:
- GPT-2: trained from datasets is collected from outbound links from Reddit (8 million docs, 40 GB)
- Training Dataset
- a random sample of the Jan 2022 snapshot of CommonCrawl. (3.3 billion tokens, 9GB)



Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Experiment

Efficiency in Data Processing

- How many hours required to compute PPL for 1 Bilion token text examples (3GB)

Models	Computation			Model Para#
	Full Batch	Half Batch	Flash-Att	
GPT-2 XL (Radford et al., 2019a)	34.57	31.68	–	1.5B
GPT-Neo 2.7B (Black et al., 2021)	48.47	43.74	–	2.7B
Llama2-7B (Touvron et al., 2023)	96.99	86.29	82.29	7B
Llama2-13B (Touvron et al., 2023)	149.65	138.11	130.57	13B

- We tried several acceleration tricks. Reduce computation batch size and use flash attention.

Experiment

Evaluating Model Performance

Methods	NLL		Commonsense Accuracy			
Datasets	C4	Wiki	winogrande	boolq	hellaswag	Average
GPT2-base	1600	2320	51.62	48.72	31.14	43.83
different parts of data						
Llama13b-head-20	1560	2272	49.33	52.14	31.11	44.2
Llama13b-middle-20	1552	2272	50.28	52.39	31.19	44.62
Llama13b-tail-20	1560	2272	51.22	51.38	31.05	44.55
Llama13b-middle-30	1552	2288	51.07	50.03	31.18	44.1
Llama13b-middle-40	1552	2288	50.59	47.22	30.97	42.93
different sizes of quality estimation models						
Llama7b-middle-20	1560	2288	51.7	53.46	31.22	45.46
GPTNeo-middle-20	1560	2288	51.85	53.49	31.2	45.52
GPT2XL-middle-20	1560	2288	51.54	52.91	31.25	45.24
Baichuan7b-middle-20	1560	2288	51.7	53.15	31.2	45.35
training dynamics						
Baichuan7b-meanmin-20	1560	2272	51.93	54.46	31.03	45.81
Baichuan7b-meanmax-20	1560	2288	51.7	54.16	31.05	45.64
Baichuan7b-varmax-20	1560	2272	51.07	54.13	31.07	45.43
train a larger model						
GPT2-large	1400	2008	55.33	60.49	45.35	53.73
Llama7b-middle-20	1376	2040	56.59	61.41	45	54.34
Llama13b-middle-20	1376	2040	55.72	61.65	44.98	54.12

NLL can not well evaluate the performance difference.

Experiment

Evaluating Model Performance

Methods	NLL		Commonsense Accuracy			
Datasets	C4	Wiki	winogrande	boolq	hellaswag	Average
GPT2-base	1600	2320	51.62	48.72	31.14	43.83
different parts of data						
Llama13b-head-20	1560	2272	49.33	52.14	31.11	44.2
Llama13b-middle-20	1552	2272	50.28	52.39	31.19	44.62
Llama13b-tail-20	1560	2272	51.22	51.38	31.05	44.55
Llama13b-middle-30	1552	2288	51.07	50.03	31.18	44.1
Llama13b-middle-40	1552	2288	50.59	47.22	30.97	42.93
different sizes of quality estimation models						
Llama7b-middle-20	1560	2288	51.7	53.46	31.22	45.46
GPTNeo-middle-20	1560	2288	51.85	53.49	31.2	45.52
GPT2XL-middle-20	1560	2288	51.54	52.91	31.25	45.24
Baichuan7b-middle-20	1560	2288	51.7	53.15	31.2	45.35
training dynamics						
Baichuan7b-meanmin-20	1560	2272	51.93	54.46	31.03	45.81
Baichuan7b-meanmax-20	1560	2288	51.7	54.16	31.05	45.64
Baichuan7b-varmax-20	1560	2272	51.07	54.13	31.07	45.43
train a larger model						
GPT2-large	1400	2008	55.33	60.49	45.35	53.73
Llama7b-middle-20	1376	2040	56.59	61.41	45	54.34
Llama13b-middle-20	1376	2040	55.72	61.65	44.98	54.12

Middle part give the best performance.

Experiment

Evaluating Model Performance

Methods	NLL		Commonsense Accuracy			
Datasets	C4	Wiki	winogrande	boolq	hellaswag	Average
GPT2-base	1600	2320	51.62	48.72	31.14	43.83
different parts of data						
Llama13b-head-20	1560	2272	49.33	52.14	31.11	44.2
Llama13b-middle-20	1552	2272	50.28	52.39	31.19	44.62
Llama13b-tail-20	1560	2272	51.22	51.38	31.05	44.55
Llama13b-middle-30	1552	2288	51.07	50.03	31.18	44.1
Llama13b-middle-40	1552	2288	50.59	47.22	30.97	42.93
different sizes of quality estimation models						
Llama7b-middle-20	1560	2288	51.7	53.46	31.22	45.46
GPTNeo-middle-20	1560	2288	51.85	53.49	31.2	45.52
GPT2XL-middle-20	1560	2288	51.54	52.91	31.25	45.24
Baichuan7b-middle-20	1560	2288	51.7	53.15	31.2	45.35
training dynamics						
Baichuan7b-meanmin-20	1560	2272	51.93	54.46	31.03	45.81
Baichuan7b-meanmax-20	1560	2288	51.7	54.16	31.05	45.64
Baichuan7b-varmax-20	1560	2272	51.07	54.13	31.07	45.43
train a larger model						
GPT2-large	1400	2008	55.33	60.49	45.35	53.73
Llama7b-middle-20	1376	2040	56.59	61.41	45	54.34
Llama13b-middle-20	1376	2040	55.72	61.65	44.98	54.12

Its not larger model gives better estimation.

Experiment

Evaluating Model Performance

Methods	NLL		Commonsense Accuracy			
Datasets	C4	Wiki	winogrande	boolq	hellaswag	Average
GPT2-base	1600	2320	51.62	48.72	31.14	43.83
different parts of data						
Llama13b-head-20	1560	2272	49.33	52.14	31.11	44.2
Llama13b-middle-20	1552	2272	50.28	52.39	31.19	44.62
Llama13b-tail-20	1560	2272	51.22	51.38	31.05	44.55
Llama13b-middle-30	1552	2288	51.07	50.03	31.18	44.1
Llama13b-middle-40	1552	2288	50.59	47.22	30.97	42.93
different sizes of quality estimation models						
Llama7b-middle-20	1560	2288	51.7	53.46	31.22	45.46
GPTNeo-middle-20	1560	2288	51.85	53.49	31.2	45.52
GPT2XL-middle-20	1560	2288	51.54	52.91	31.25	45.24
Baichuan7b-middle-20	1560	2288	51.7	53.15	31.2	45.35
training dynamics						
Baichuan7b-meanmin-20	1560	2272	51.93	54.46	31.03	45.81
Baichuan7b-meanmax-20	1560	2288	51.7	54.16	31.05	45.64
Baichuan7b-varmax-20	1560	2272	51.07	54.13	31.07	45.43
train a larger model						
GPT2-large	1400	2008	55.33	60.49	45.35	53.73
Llama7b-middle-20	1376	2040	56.59	61.41	45	54.34
Llama13b-middle-20	1376	2040	55.72	61.65	44.98	54.12

Training dynamics can give better estimations.

Conclusion: Limitations and Takeaways

Limitations

Recognizing Constraints and Future Research

- Our research is still limited in size and need more time to explore the details.
- We need to train a **larger model** with **more data examples**.

Conclusion

Summarizing Key Takeaways

- Estimating data quality with model based metrics is time consuming.
- Training dynamics can provide a better estimation for data quality.
- Selecting data in the middle can boost model performance and stabilize training by addressing data complexity.