The IaaS Concept of DATA4LLM

https://github.com/weAIDB/awesome-data-llm

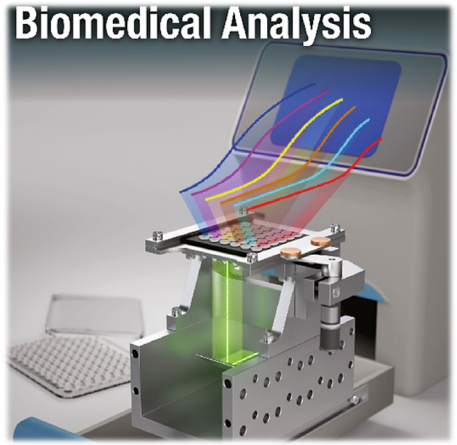
https://arxiv.org/pdf/2505.18458

Data: The Fuel of LLM Development

➢ LLMs have been widely used, which require **vast, high-quality** data, distributing across **diverse** domains

LLM Applications

General Domains



Specific Domains

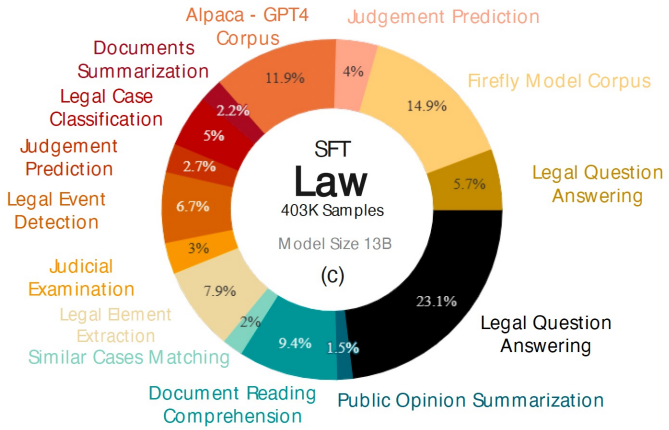
LLM Datasets

Qwen-2VL → *Qwen-2.5VL*

1.2 → **4 trillion** tokens

(with over **10,000** object categories for visual grounding)

DISC-Law-SFT



Data Size ↑

Data Quality ↑

A Survey of LLM × DATA. https://github.com/weAIDB/awsome-data-llm 🔥

Question to Ask:

*What Makes a **Dataset** Truly "Good" for LLMs?*

Inclusiveness

Abundance

Articulation

Sanitization

Data: The Fuel of LLM Development

➢ Different LLM stages have divese dataset requirements

1. Pre-Training

- 1T+ unlabeled samples

2. Continual Pre-Training

- 10M+ unlabeled samples

3. Supervised Finetuning

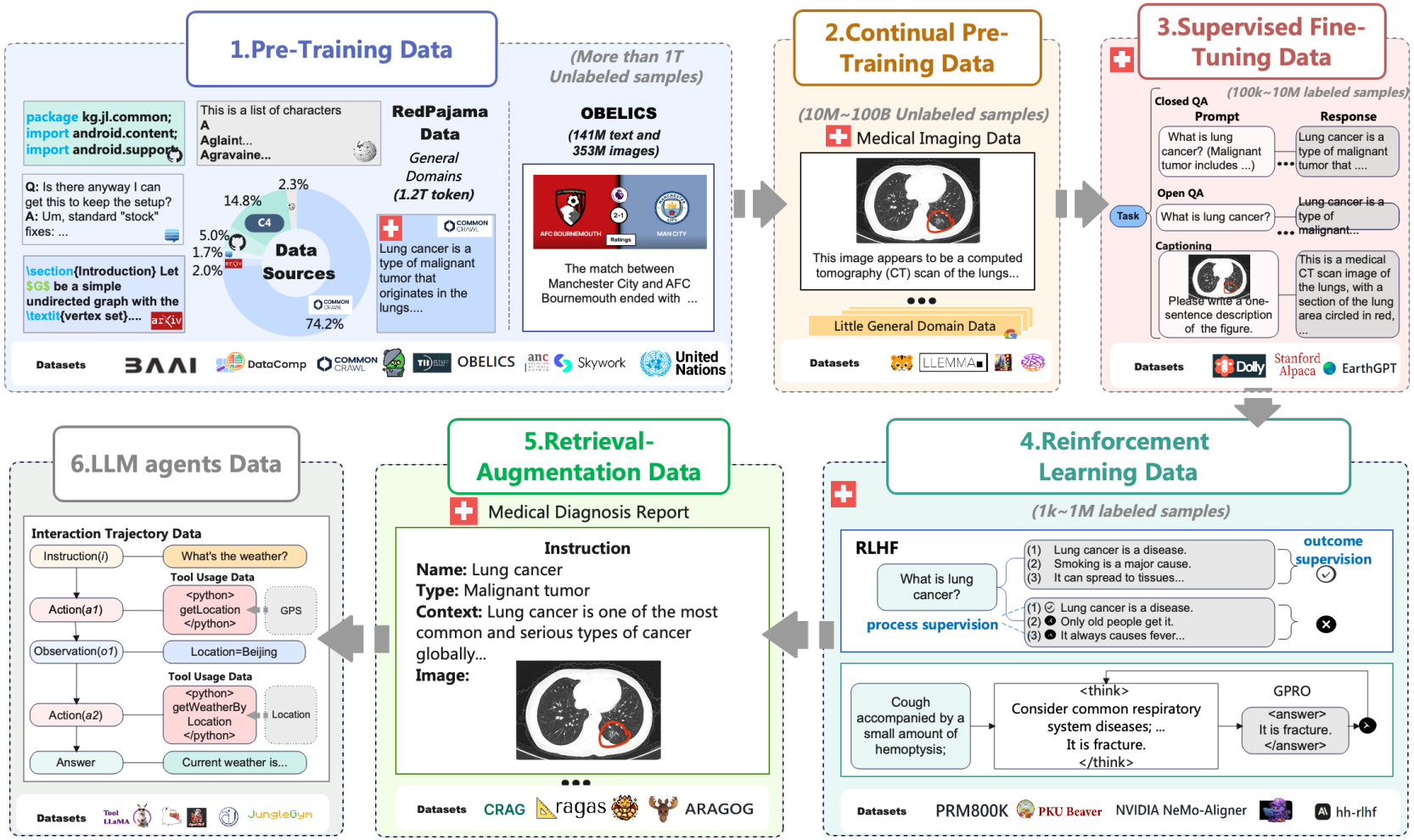
- 10k+ labeled samples

4. Reinforcement Learning

- 1k+ labeled samples

5. RAG

6. Agentic LLM

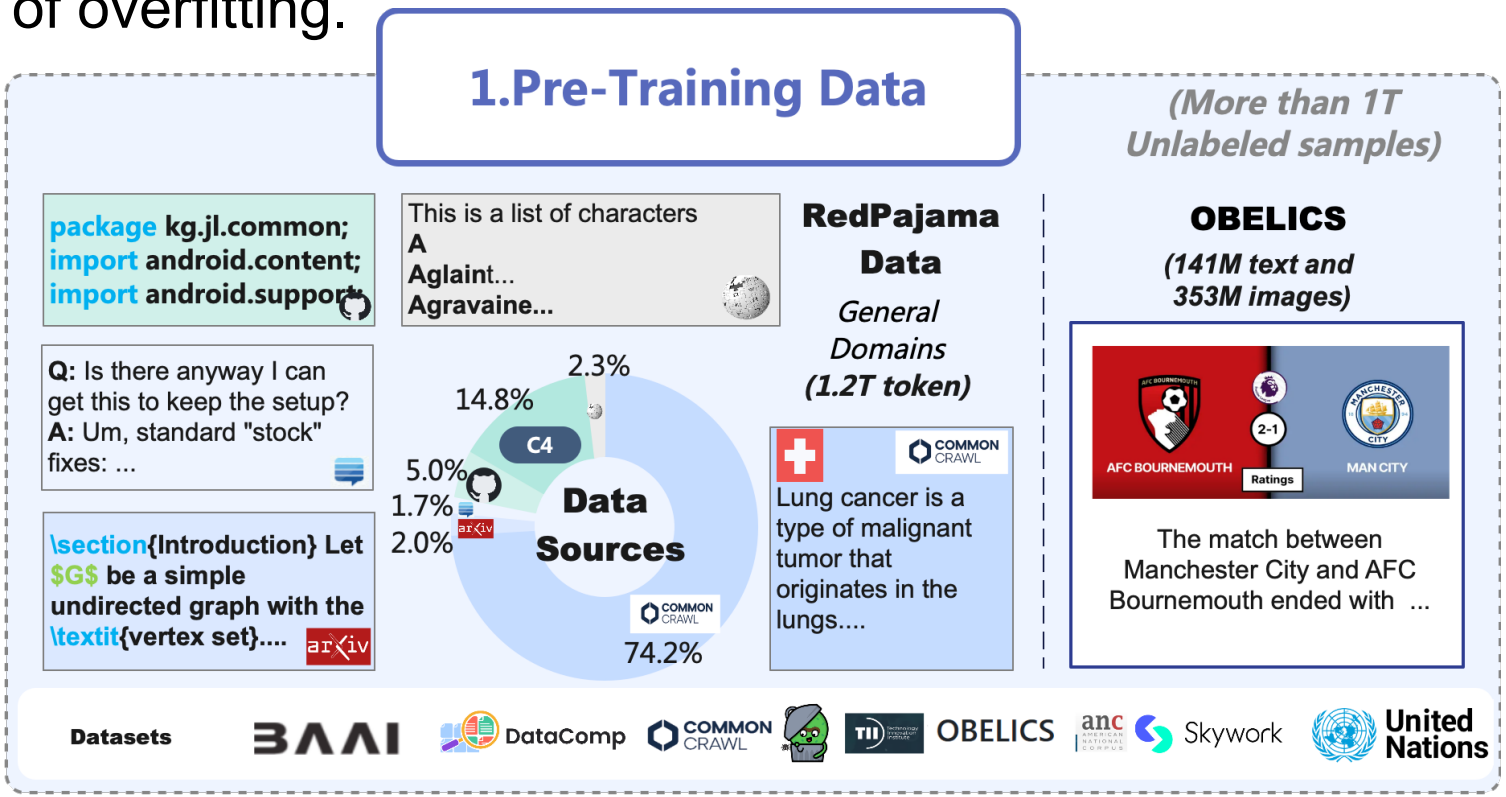


Data: The Fuel of LLM Development

➢ Different LLM stages have divese dataset requirements

1. Pre-Training (1T+ unlabeled samples)

- **Data Demand:** 1) Acquire broad language (and cross-modality) understanding; 2) Reduce the risk of overfitting.
- **Sources**
- 1) Web crawls (e.g., HTML, WARC)
- 2) Open code repository
- 3) Books (text, EPUB)
- 4) Academic papers
- 5) Interleaved image-text

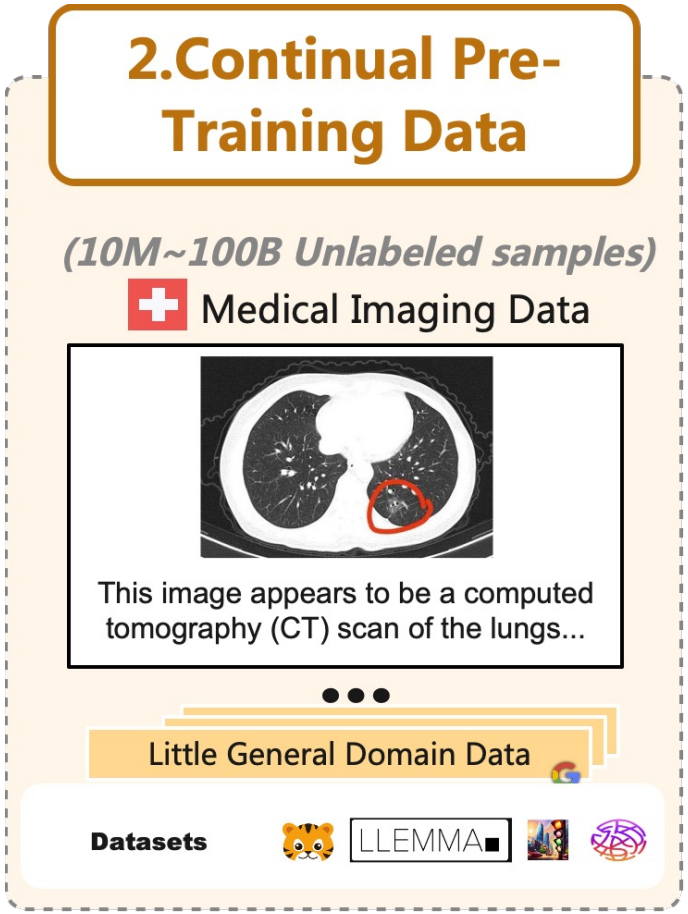


Data: The Fuel of LLM Development

➢ Different LLM stages have divese dataset requirements

2. Continual Pre-Training (10M+ unlabeled samples)

- **Data Demand:** 1) Fill knowledge gaps and 2) adapt the model to specific domains
- **Examples:**
- 1) BBT-FinCorpus: 300 GB of finance data
- 2) Medical-pt: 360,000 Chinese-English entries from medical encyclopedias



Data: The Fuel of LLM Development

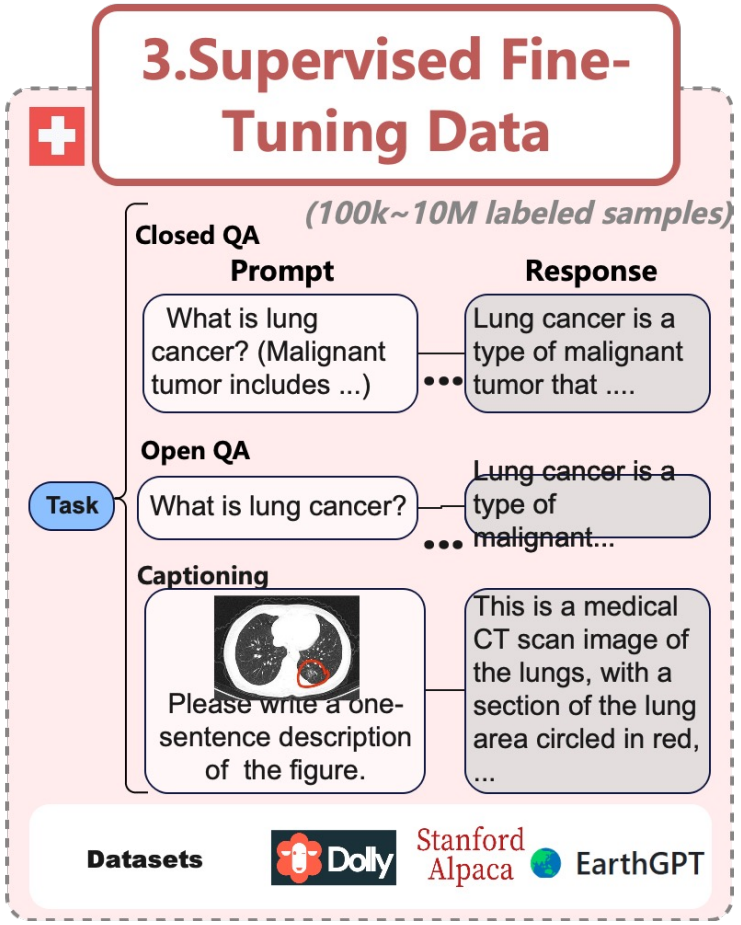
➢ Different LLM stages have divese dataset requirements

3. Supervised Finetuning (10k+ labeled samples)

- **Data Demand:** Guide the model in learning a specific, narrower set of tasks
- **General Instruction Following**
- Databricks-dolly-15K with 7 types of tasks, e.g.,

closed QA, summarization, information extraction

- **Specific Task Resolving**
- DISC-Law-SFT: Legal information extraction (32k), legal judgment prediction (16k), legal event detection (27k), and legal question-answering (93k)

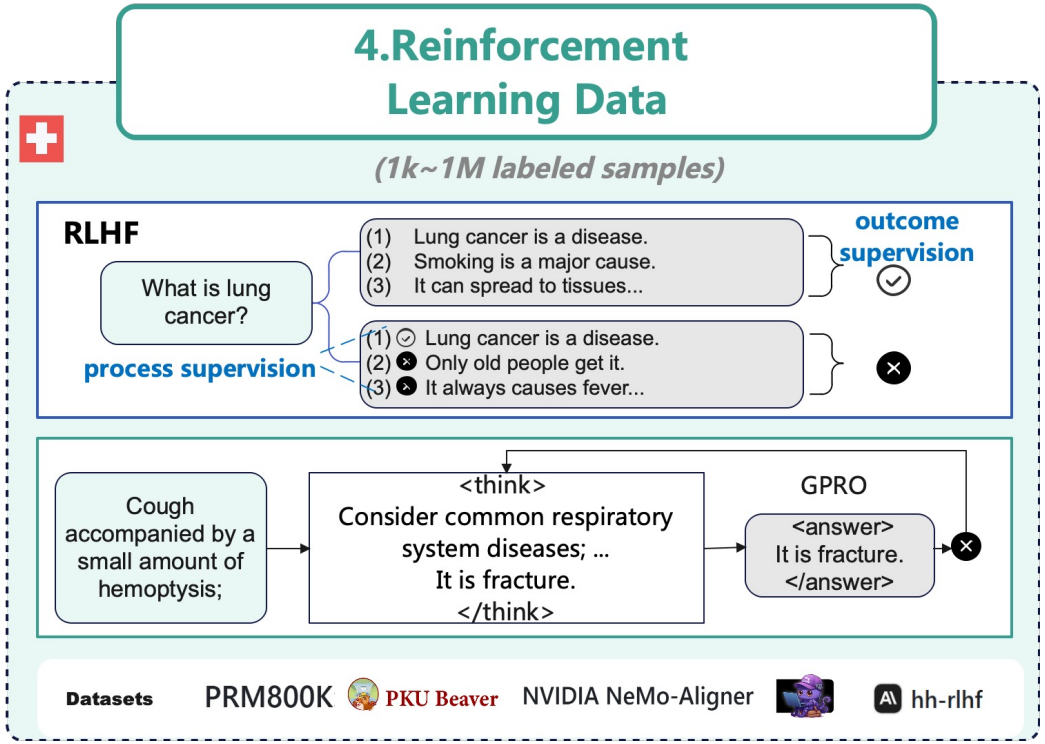


Data: The Fuel of LLM Development

➢ Different LLM stages have divese dataset requirements

4. Reinforcement Learning (1k+ labeled samples)

- **RLHF (Reinforcement Learning with Human Feedback)**
- **Data:** Smaller than SFT's with more complex data annotations (compare and rank multiple candidate responses by *human preference*)
- **Reasoning-oriented Reinforcement Learning (RoRL)**
- **Data:** Only feedback on whether the answer is correct or not (for long-term reasoning)

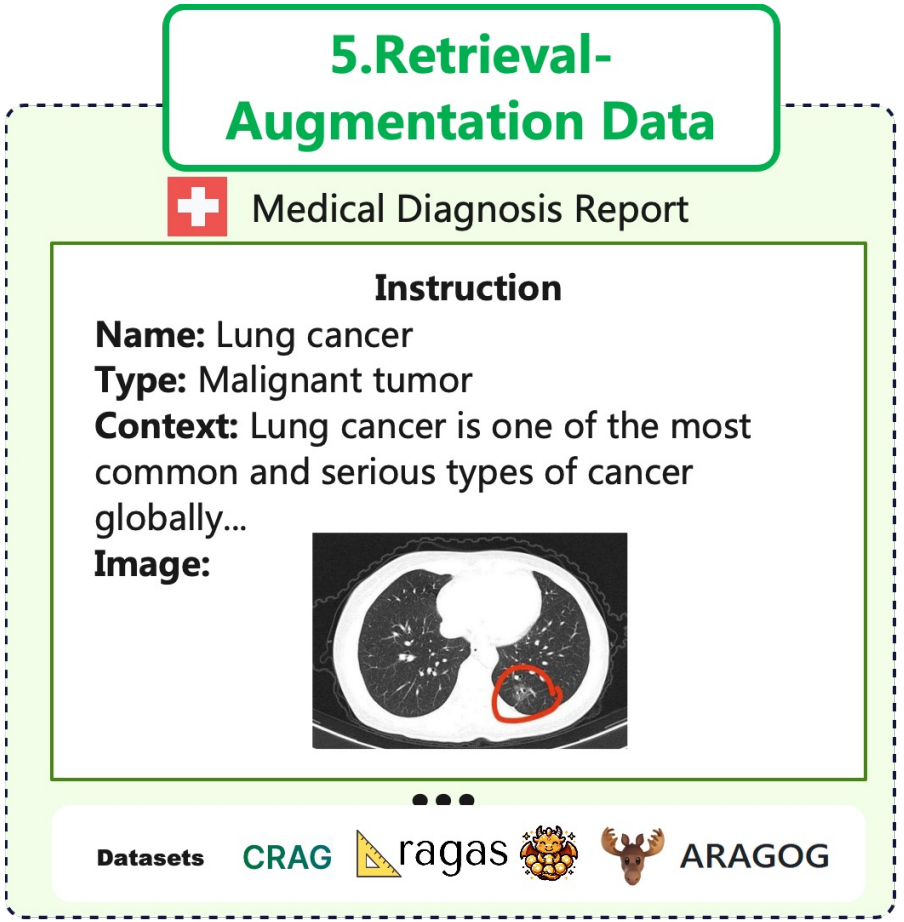


Data: The Fuel of LLM Development

➢ Different LLM stages have divese dataset requirements

5. RAG

- **Data Demand:** Strictly reviewed to ensure authenticity and validity, while dynamic data requires real-time updates
- **Example**
- **Medical:** Data from over 65,000 ICU patients and more than 200,000 patients treated in emergency department
- **Legal:** 800+ national and local laws, regulations, and rules, aswell as 24,000 legal-related exam questions.
- **User-personalized LLM**

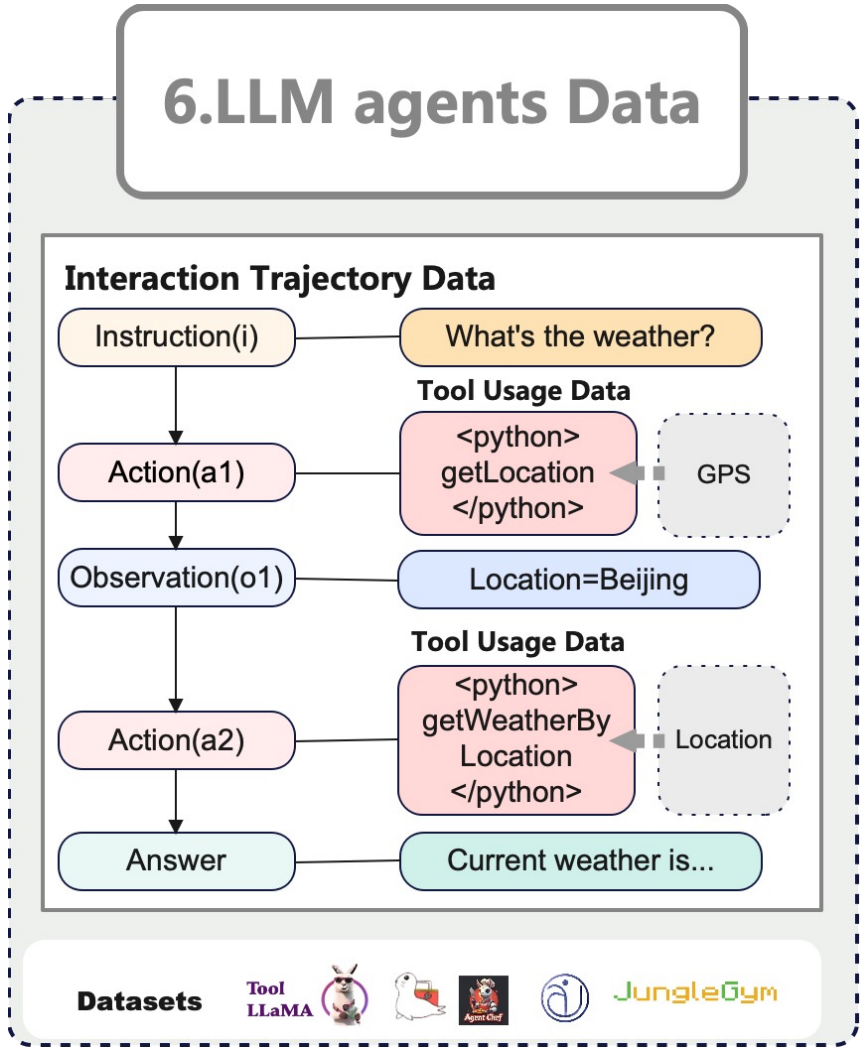


Data: The Fuel of LLM Development

➢ Different LLM stages have divese dataset requirements

6. Agent

- **Data Demand:** Require training data for advanced capabilities such as planning, tool orchestration, and multi-turn dialogue capability
- **Example**
- **Reasoning (UltraInteract):** Instruction as the root node; Both the correct actions and corresponding incorrect actions as nodes to construct a trajectory tree of human preference
- **Tool (AutoTools):** Finetune on tool data (<python>code</python>)
- **Dialogue (UltraChat):** Add an LLM to simulate user instructions and conversational content

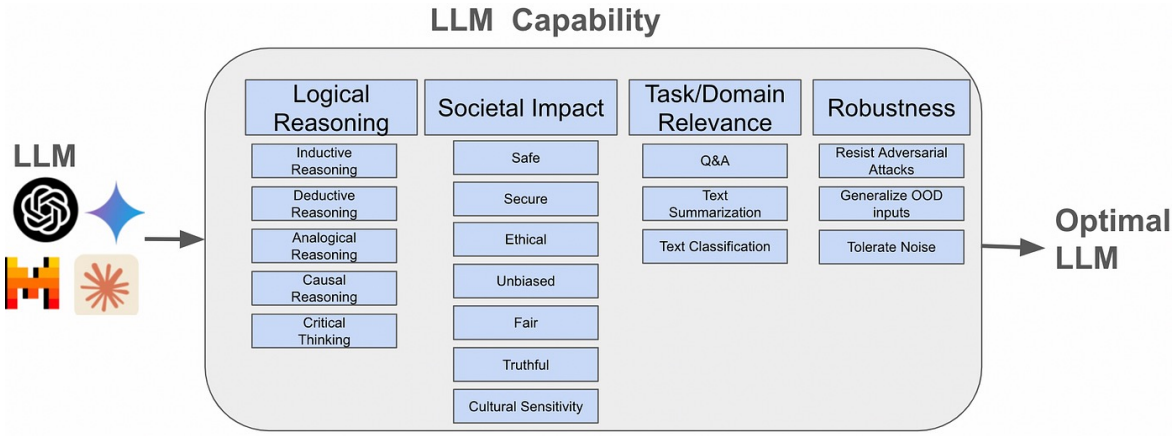


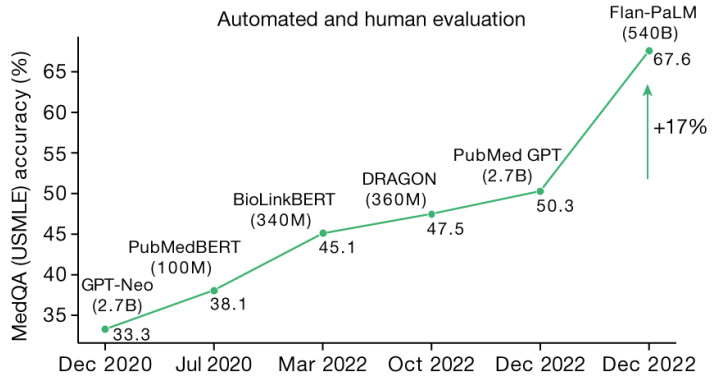
Data: The Fuel of LLM Development

➢ Different LLM stages have divese dataset requirements

7. Evaluation

- **Data:** Representative data samples that reflect different aspects of an LLM's capabilities.
- **General Domain**
- MMMU: Perception, knowledge, and reasoning
- **Specific Domain**
- HumanEval: 164 programming problems
- MedQA: 61,097 medical exam questions from various regions





Q: How long does it take for newborn jaundice to go away?

Med-PaLM: Newborn jaundice is when a newborn baby's skin and eyes look yellow. It is caused by a buildup of a substance called bilirubin in the baby's blood. Bilirubin is a byproduct of the normal process of red blood cells breaking down. Newborn jaundice is common and typically harmless, but it can be a sign of a more serious condition. The jaundice typically goes away on its own within a few weeks. In some cases, it may last longer or require treatment. The time it takes for the jaundice to go away can vary depending on the cause and the severity of the jaundice. If the jaundice is severe or lasts longer than a few weeks, the doctor may recommend testing or treatment to determine the cause and prevent complications.

The IaaS Concept of DATA4LLM

➢ The ***IaaS*** Concept

1. Inclusiveness

- Domains; Task Types; Sources; Languages;
- Expression Styles; Data Modalities

2. Abundance

- Enhance domain-specific ability

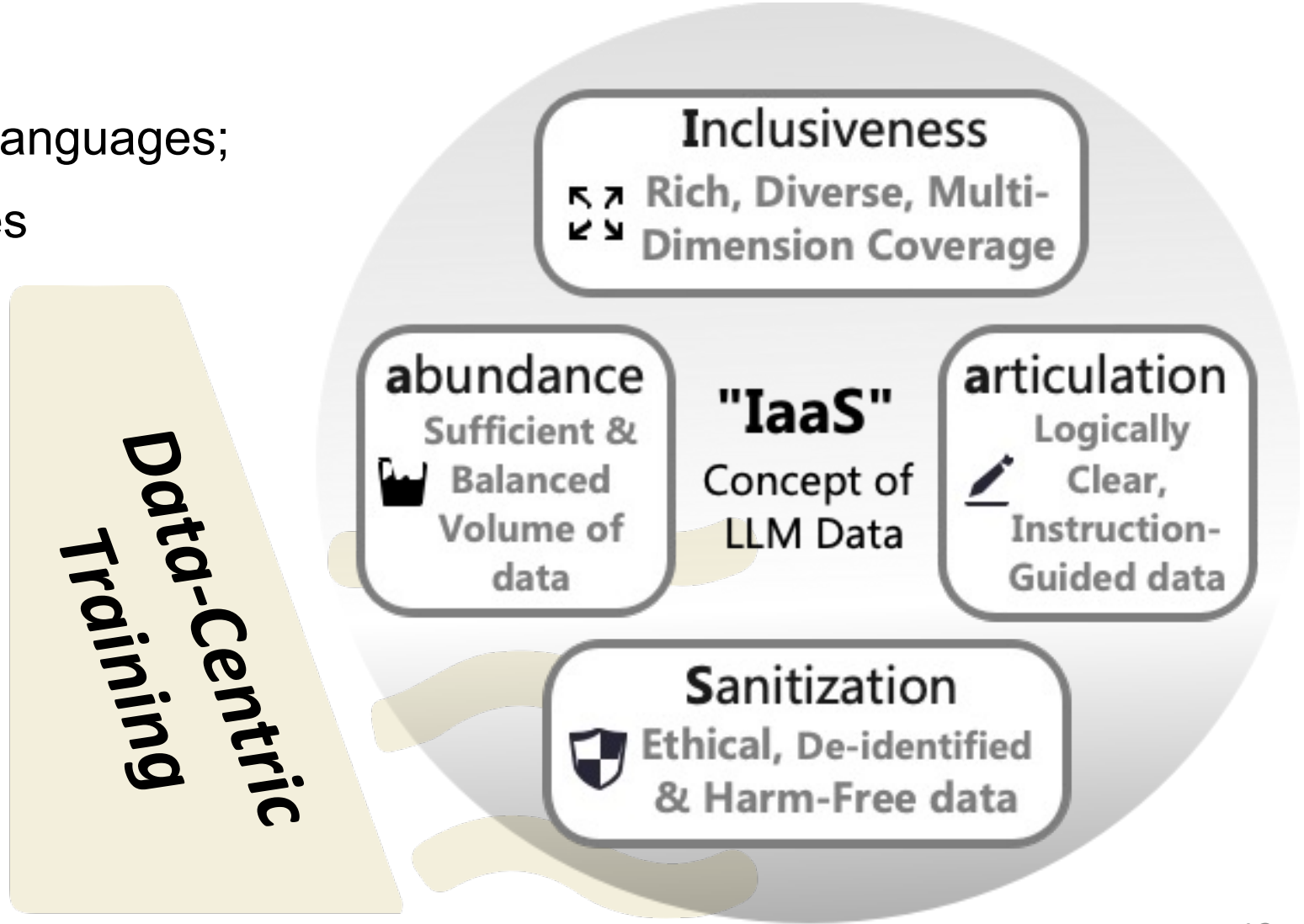
3. Articulation

- Well-Formatted; Instructive;
- Step-by-step Reasoning

4. Sanitization

- Privacy; Ethical; Risk Removal

5. Data Application Strategy

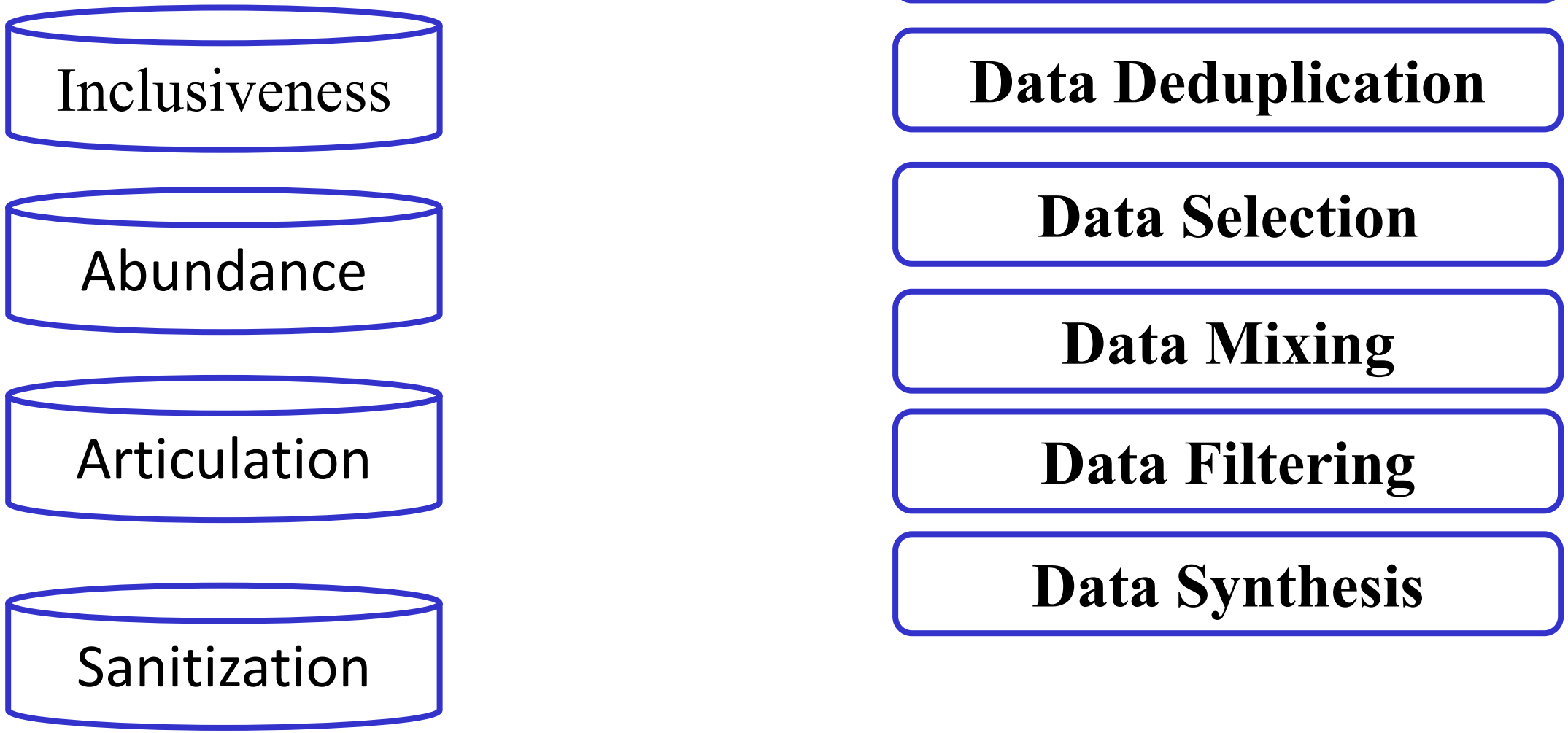


The IaaS Concept of DATA4LLM

Characters \ Stages	Pre-Training	Continual Pre-Training	Finetuning	RL	RAG	Agent	Evaluation
Inclusiveness	**Diverse Domains:** General Knowledge, Coding, STEM, Multilingual ...	**Specific Domain:** Finance, Medicine, ...	**Specific Domain:** QA, Summarization, Writing, Classification...	**Specific Domain:** **RLHF:** General, Medicine, Law... **RoRL:** Math, Coding, ...	**Specific Domain:** Finance, Medicine, Law...	**Planning; Tool; Multi-Turn Dialogue**	**Domain-Specific:** Legal Consultation, Code Synthesis and Translation...
Abundance	**Over 1 TB**; Tend to Increase Ratios like Math and Coding	**10M-100B**; Need General Data	**10k-10M**	**1k-1M**	---	**10k-10M**	**0.1k-100k**
Articulation	**Format:** Clear, Fluent, and Unambiguous Language; Well Formatted	**Format:** Clear, Fluent, and Unambiguous Language; Well Formatted	**Instruction-Response Pairs;** **Format:** Task-Oriented Response	**Step-By-Step Reasoning;** **Fine-Grained Annotation**	**Real-time;** **Extensive Info**	**Reasoning;** **Interaction Format**: Insturction+ Action+ Observation...	**Granular Evaluation;** **Free from Data Leakage**
Sanitization	Privacy Compliance; Toxicity-Free; Ethical Consistency; Risk Mitigation						

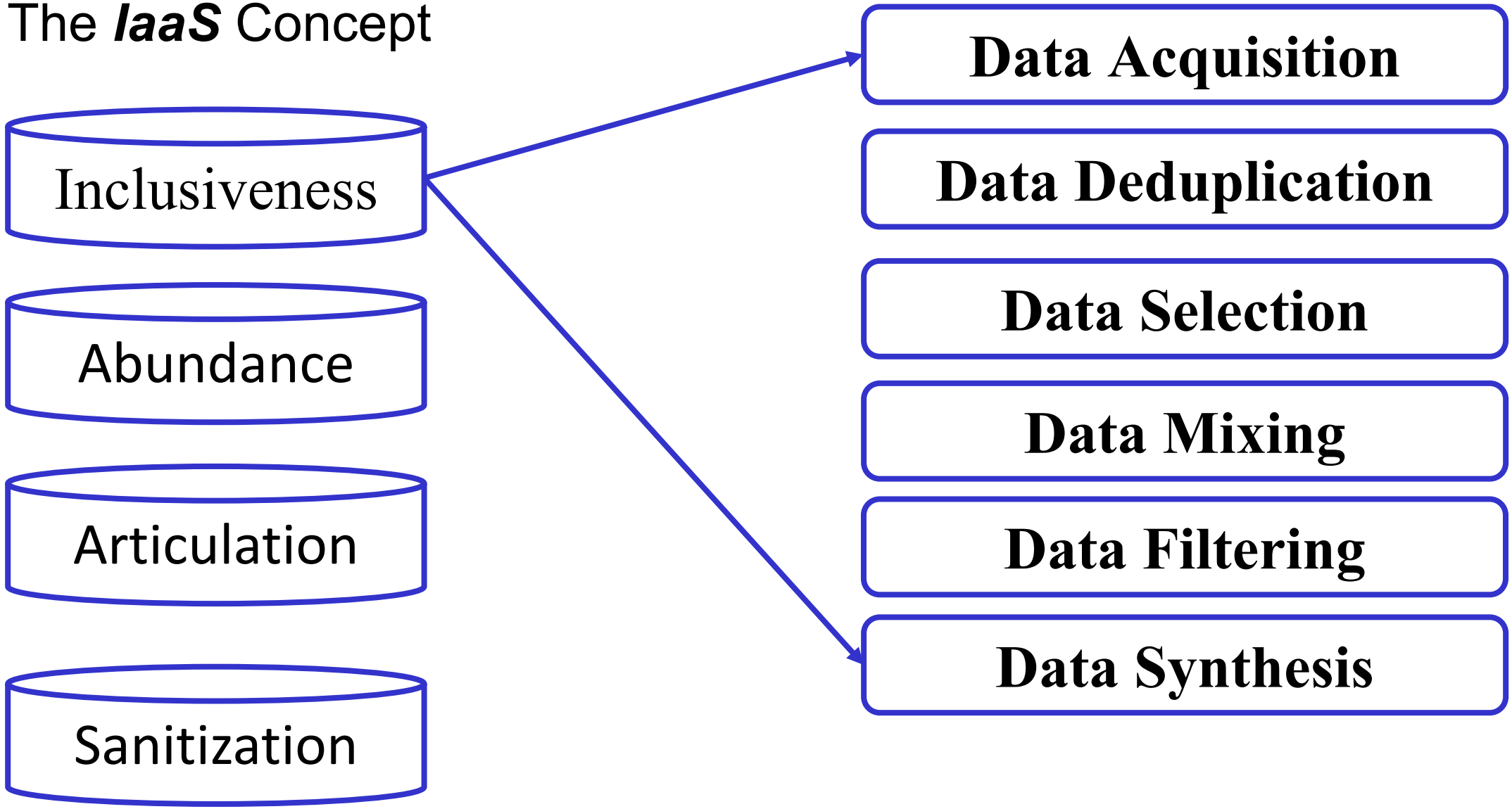
The IaaS Concept of DATA4LLM

- The ***IaaS*** Concept



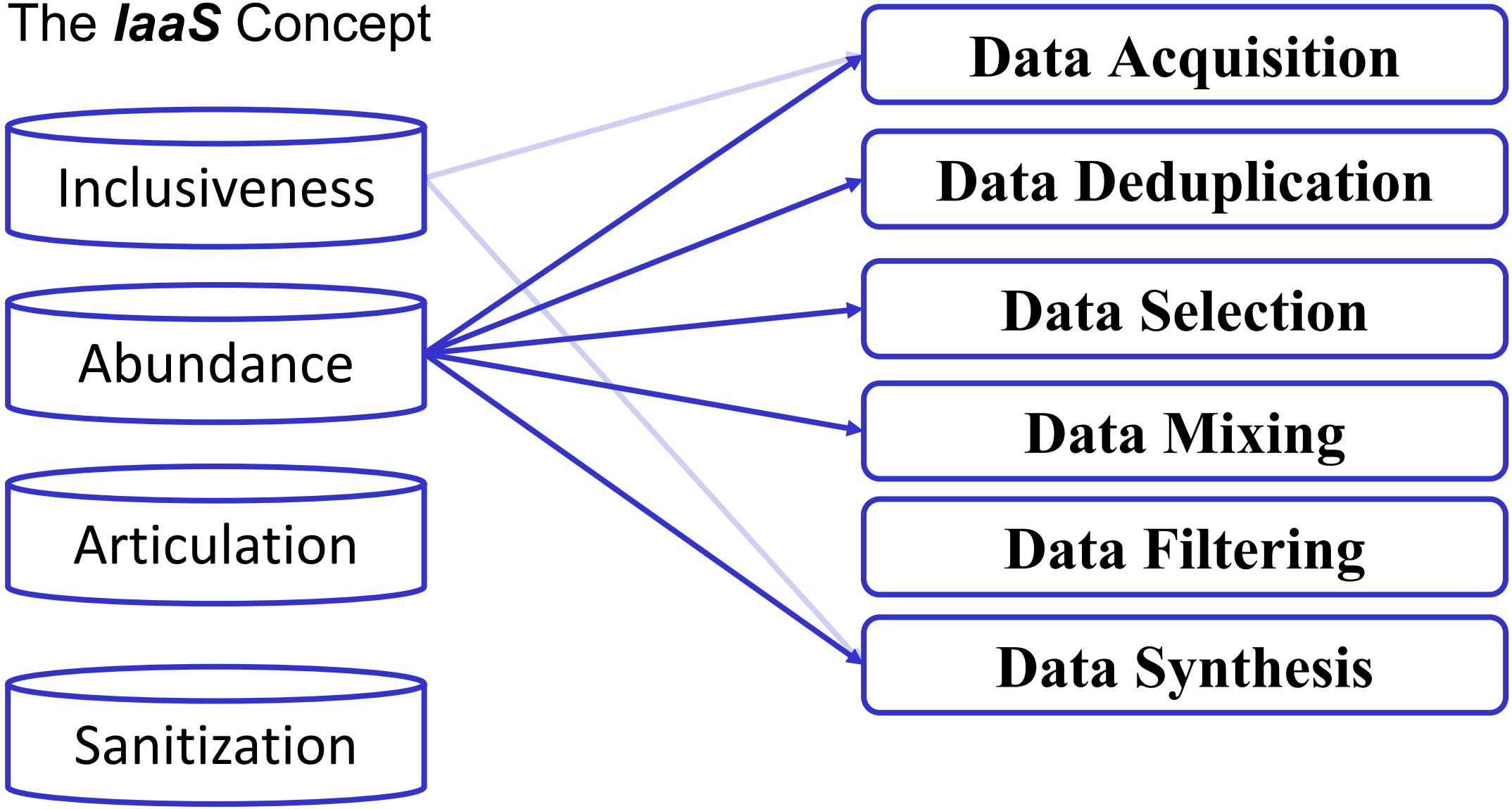
The IaaS Concept of DATA4LLM

- The ***IaaS*** Concept



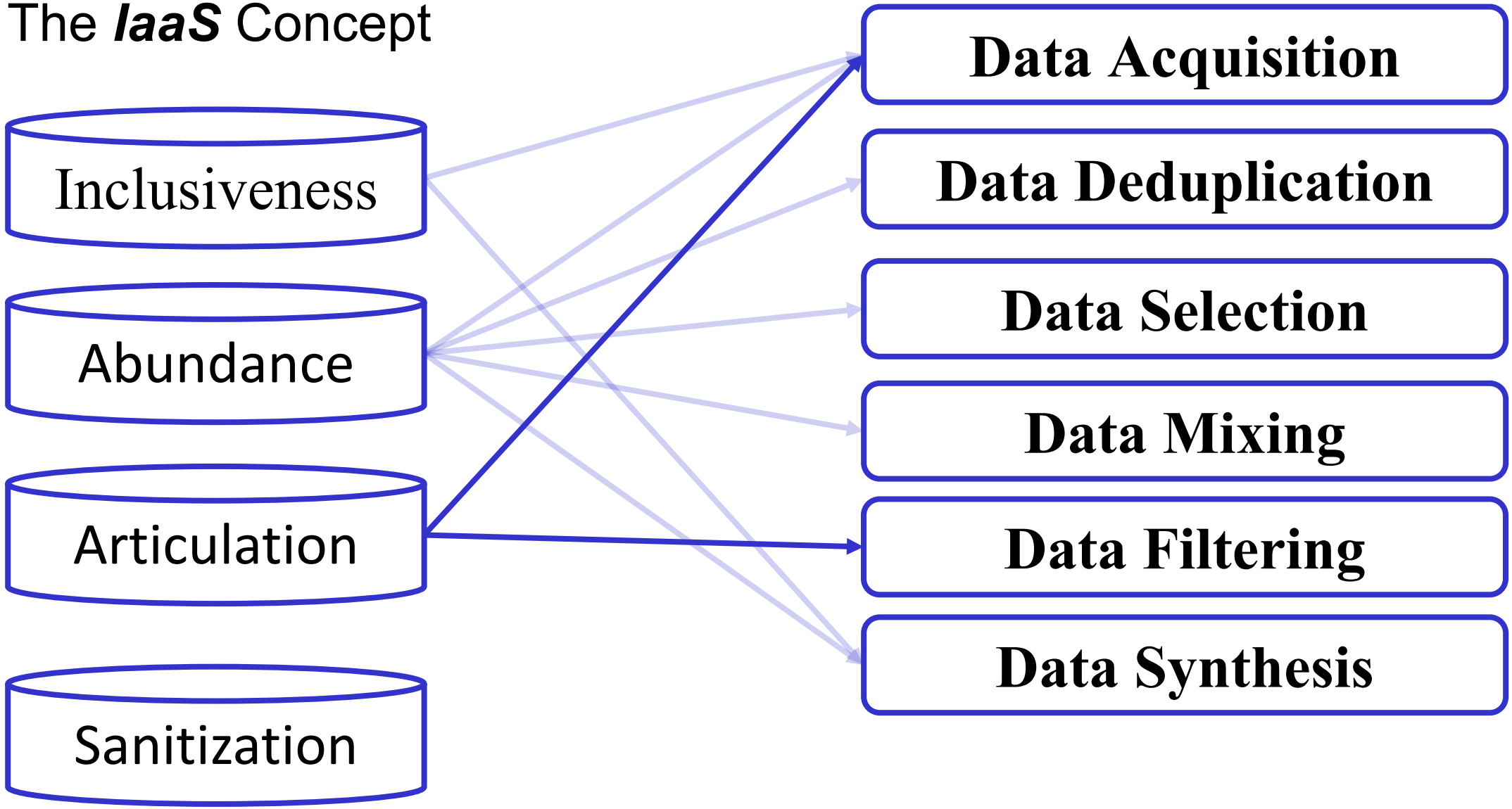
The IaaS Concept of DATA4LLM

- The ***IaaS*** Concept



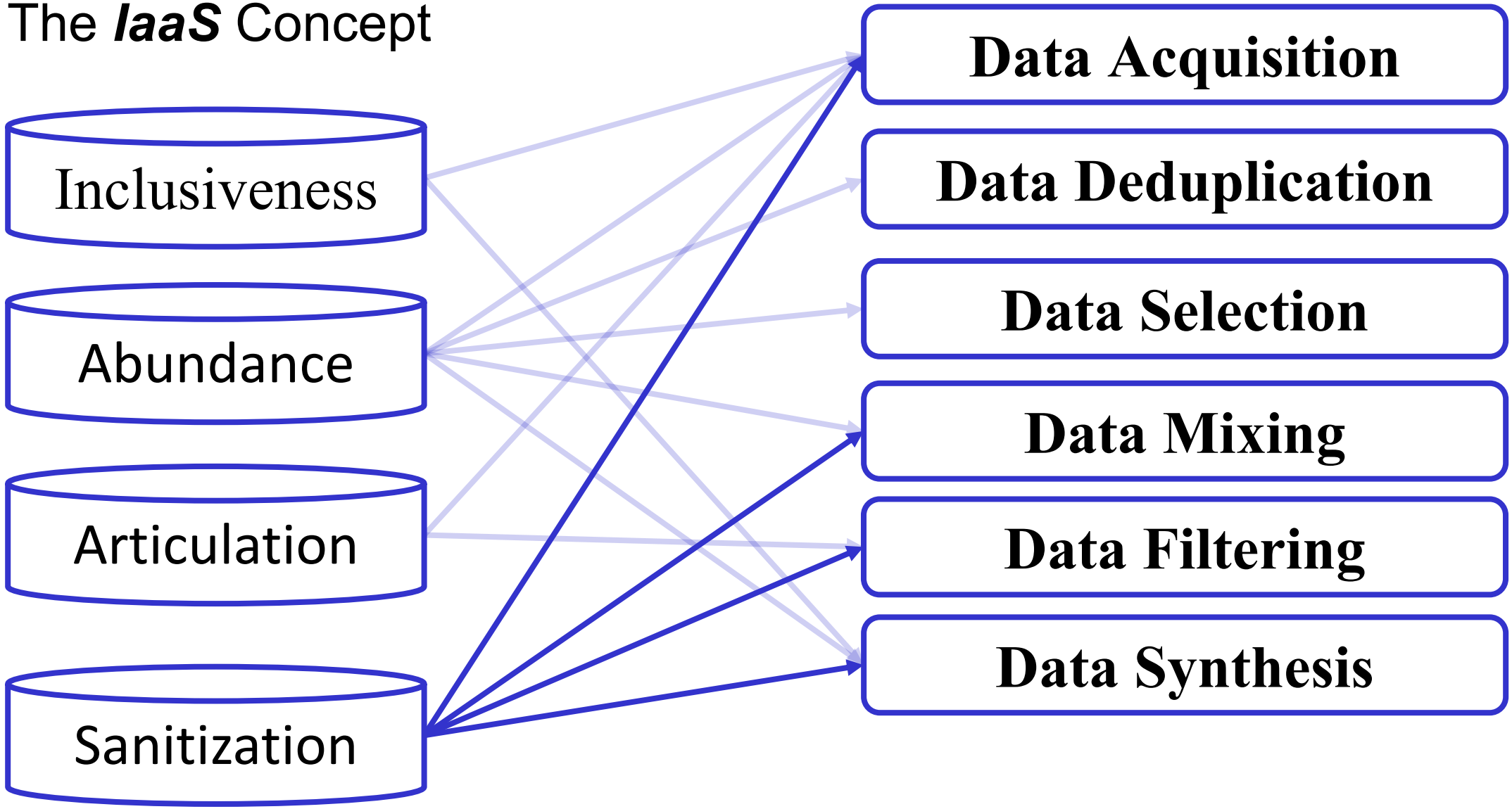
The IaaS Concept of DATA4LLM

- The ***IaaS*** Concept



The IaaS Concept of DATA4LLM

- The ***IaaS*** Concept



The IaaS Concept of DATA4LLM

➢ Required Techniques under ***IaaS***

Stage		Pre-training / Incremental Pre-training	Supervised Fine-Tuning	Reinforcement Learning	Inference	RAG
Data Processing	Acquisition	✓	✓	✓	N/A	✓
	De-duplication	✓	✓	N/A	N/A	N/A
	Filtering	✓	✓	N/A	N/A	×
	Selection	✓	✓	N/A	N/A	N/A
	Mixing	✓	✓	×	N/A	×
	Synthesis	✓	✓	✓	N/A	✓
Data Storage	Distribution	Distributed File System Model Offload (GPUs, CPUs)	Model Offload (GPUs, CPUs)	Model Offload (GPUs, CPUs)	Model Offload (GPUs, CPUs)	Model Offload (GPUs, CPUs)
	Transmission	Caching Data Placement Parallelized Pipeline Data/Operator Offloading (CPUs)	Parallelized Pipeline Data/Operator Offloading (CPUs)	Parallelized Pipeline Data/Operator Offloading (CPUs)	×	N/A
	Fault Tolerance	✓	✓	✓	×	×
	KV Cache	N/A	N/A	N/A	Cache Space Management KV Indexing KV Placement KV Shrinking	KV Placement KV Shrinking
Data Serving	Selection	Sample-Scoring-Based Model-State-Based	Model-State-Based Experience-Based	N/A	×	SLM-Based Filtering LLM-Based Filtering Metric-Based Re-ranking LLM-Based Re-ranking
	Compression	N/A	N/A	N/A	✓	✓
	Packing	✓	✓	✓	×	×
	Provenance	×	×	×	✓	N/A

The IaaS Concept of DATA4LLM

➢ Required Techniques under ***IaaS***

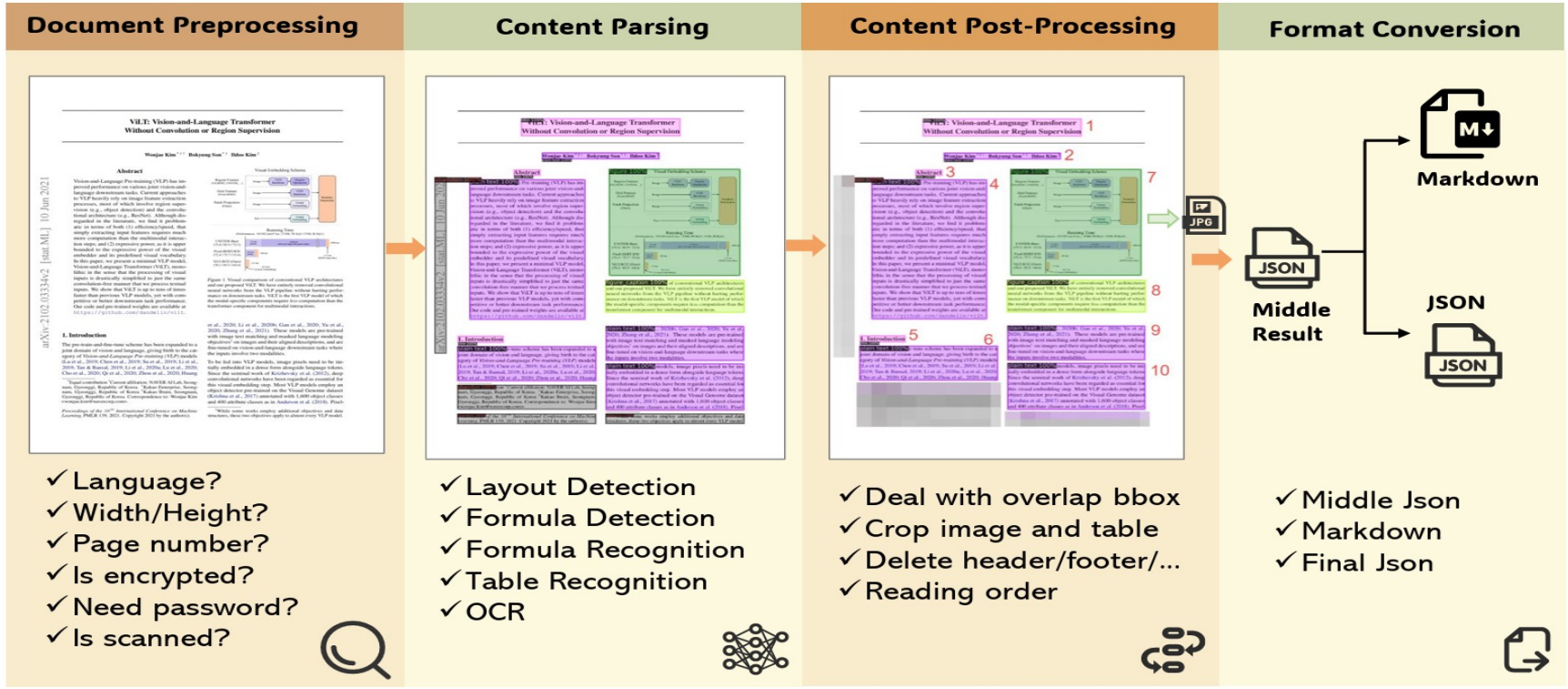
Stage		Pre-training / Incremental Pre-training	Supervised Fine-Tuning	Reinforcement Learning	Inference	RAG
Data Processing	Acquisition	✓	✓	✓	N/A	✓
	De-duplication	✓	✓	N/A	N/A	N/A
	Filtering	✓	✓	N/A	N/A	×
	Selection	✓	✓	N/A	N/A	N/A
	Mixing	✓	✓	×	N/A	×
	Synthesis	✓	✓	✓	N/A	✓
Data Storage	Distribution	Distributed File System Model Offload (GPUs, CPUs)	Model Offload (GPUs, CPUs)	Model Offload (GPUs, CPUs)	Model Offload (GPUs, CPUs)	Model Offload (GPUs, CPUs)
	Transmission	Caching Data Placement Parallelized Pipeline Data/Operator Offloading (CPUs)	Parallelized Pipeline Data/Operator Offloading (CPUs)	Parallelized Pipeline Data/Operator Offloading (CPUs)	×	N/A
	Fault Tolerance	✓	✓	✓	×	×
	KV Cache	N/A	N/A	N/A	Cache Space Management KV Indexing KV Placement KV Shrinking	KV Placement KV Shrinking
Data Serving	Selection	Sample-Scoring-Based Model-State-Based	Model-State-Based Experience-Based	N/A	×	SLM-Based Filtering LLM-Based Filtering Metric-Based Re-ranking LLM-Based Re-ranking
	Compression	N/A	N/A	N/A	✓	✓
	Packing	✓	✓	✓	×	×
	Provenance	×	×	×	✓	N/A

Data Acquisition

Method	Objective	Solution	Tools
Website Crawling	HTML Textual Content Extraction	Rule-based	Trafilatura [73]
		Rule-based	BET [144]
		ML-based	Dragnet [313]
	Automate Browser Interactions	HTML parsing	Beautiful Soup [6]
		Control web driver	Selenium [19]
		Wrap high-level API	Playwright [30]
		DevTools protocol	Puppeteer [31]
Layout-based	Content Extraction from Handwritten or Non-text Data	Model pipeline	PaddleOCR
		Model pipeline	MinerU [392]
		Multimodal LLM	GOT2.0 [407]
		Multimodal LLM	Fox [257]
Entity recognition & linking	New Sample Derivation	Bi-Transformer	ReFinED [68]
	Translation Consistency	Seq2seq Framework using References	AACTRANS [215]
	Text-Image Integration	Multimodal LLM	UMIE [367]

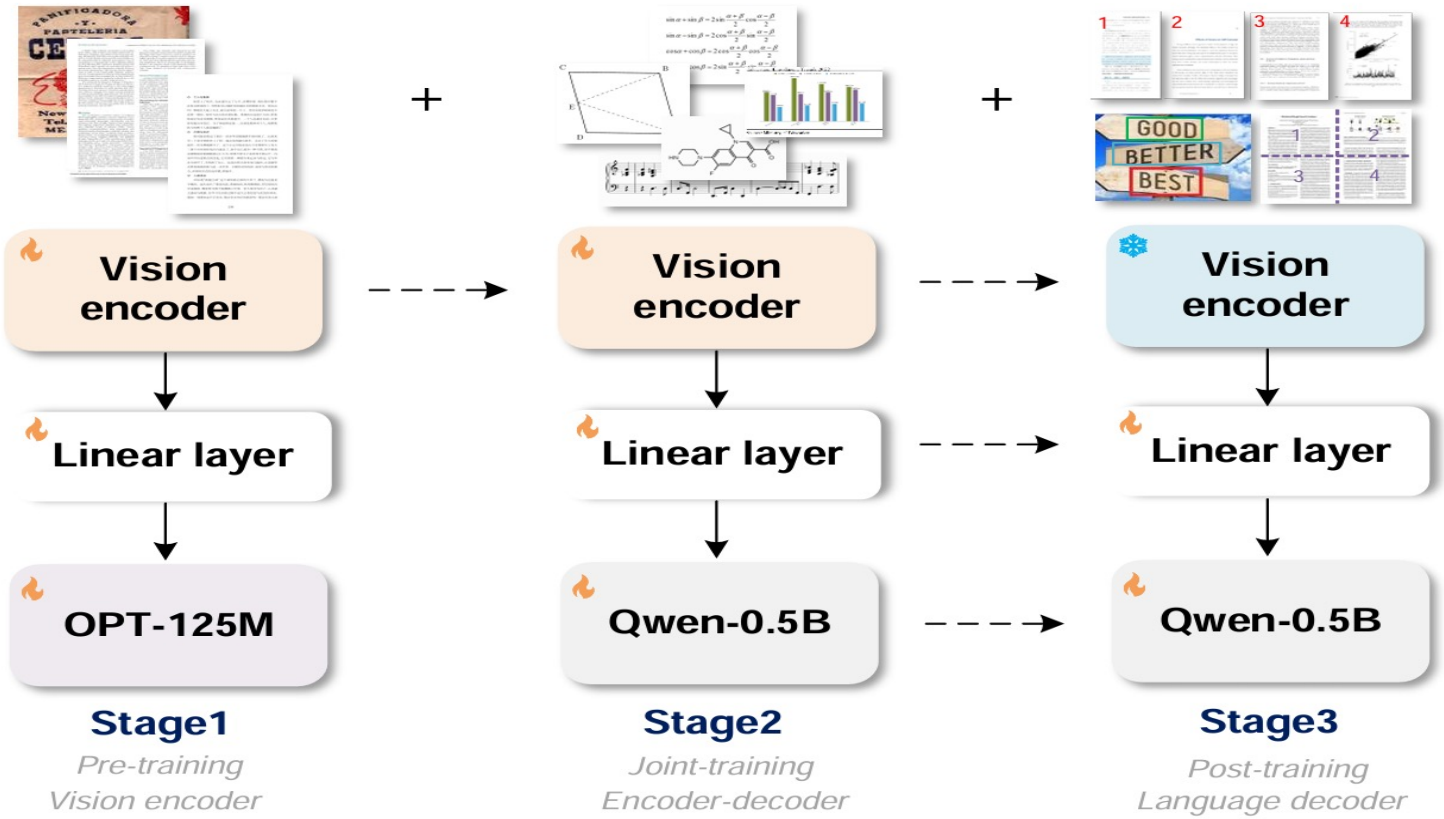
Data Acquisition —— Layout Analysis

- Challenge: How to handle various types of documents with different elements.
  - Model Pipeline: Converts raw data (e.g., scanned books) into machine-readable formats in a pipeline manner, which consist of multiple small models.



Data Acquisition —— Layout Analysis

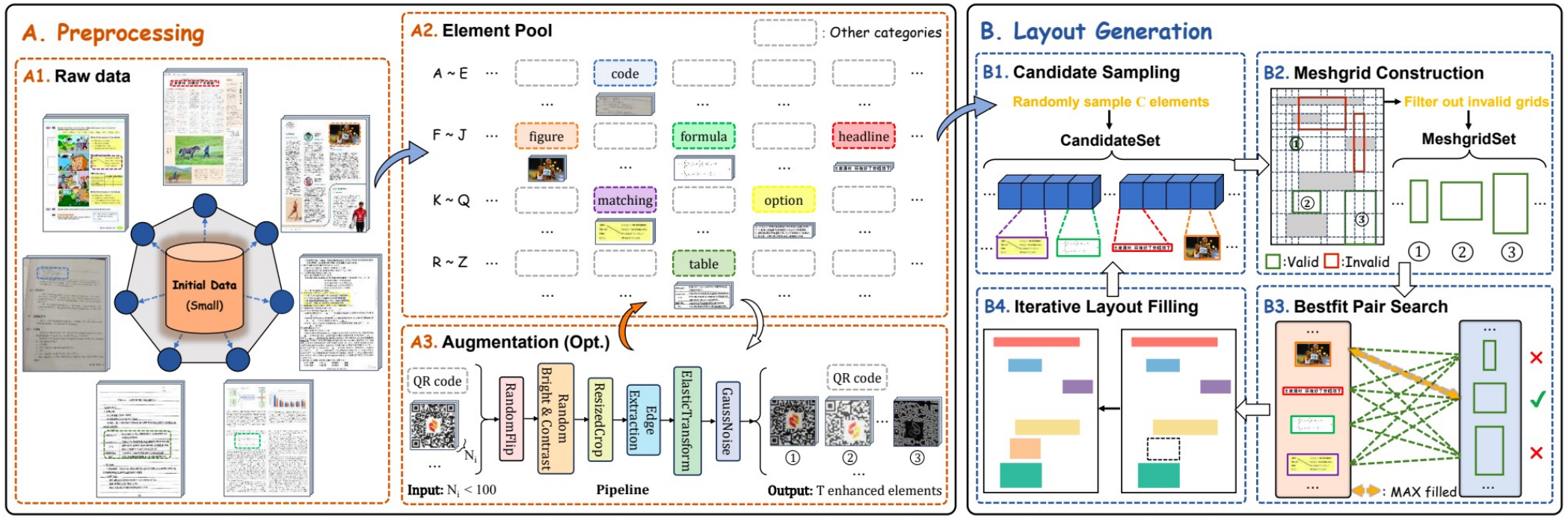
- Challenge: How to handle various types of documents with different elements.
  - Multimodal LLM: Adopts multimodal LLMs for end-to-end text acquisition.



Haoran Wei, et al. General OCR Theory: Towards OCR-2.0 via a Unified End-to-end Model[J]. arXiv preprint arXiv:2409.01704, 2024

Data Acquisition —— Layout Analysis

- Challenge: Existing dataset for layout analysis has limited layout types and volume.
  - Generate diverse document images by searching for the best match between candidate elements (Candidates) and idle blocks (Mesh).



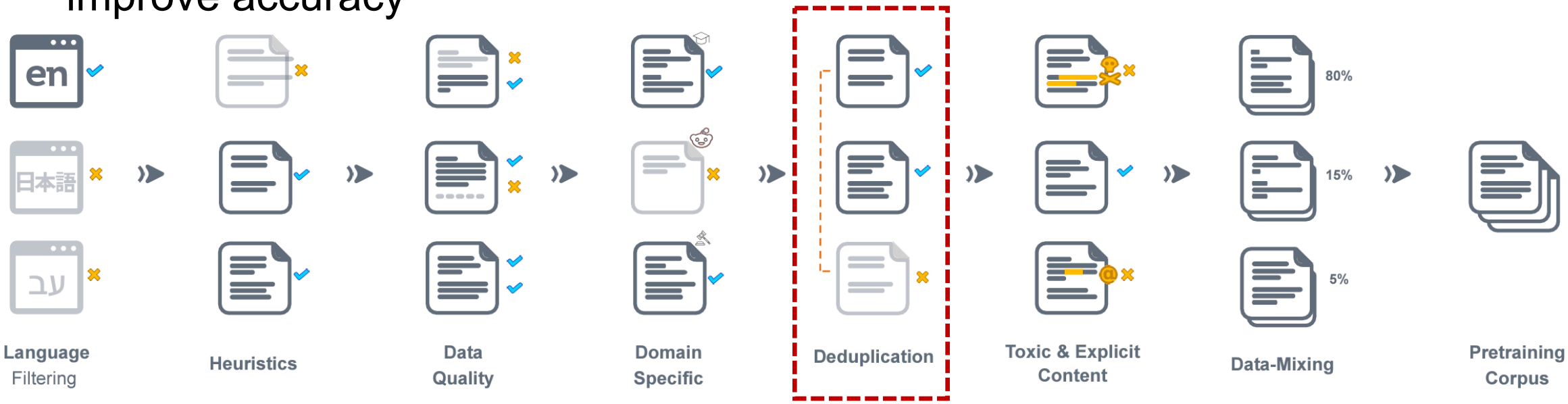
Zhiyuan Zhao, et al. DocLayout-YOLO: Enhancing Document Layout Analysis through Diverse Synthetic Data and Global-to-Local Adaptive Perception[J]. arXiv preprint arXiv:2410.12628, 2024

Data Deduplication

Method	Objective	Modality	Work
Exact substring matching	Deduplicate samples with identical substrings	Text	MD5 [122] Suffix Array [299]
Hashing identification	Deduplicate samples with similar substrings	Text	SimHash [88] MinHash [81], [122], [299] MinHashLSH [347], [358] MinHash + Bloom Filter [207] DotHash [298]
Frequency analysis	Down-weighing samples with higher commonness	Text	SoftDeDup [167]
Embedding-based clustering	Deduplicate samples with identical topics but different formats	Text + Image	SemDeDup [46] SemDeDup + SSL Prototypes[385] FairDeDup [360]

Data Deduplication

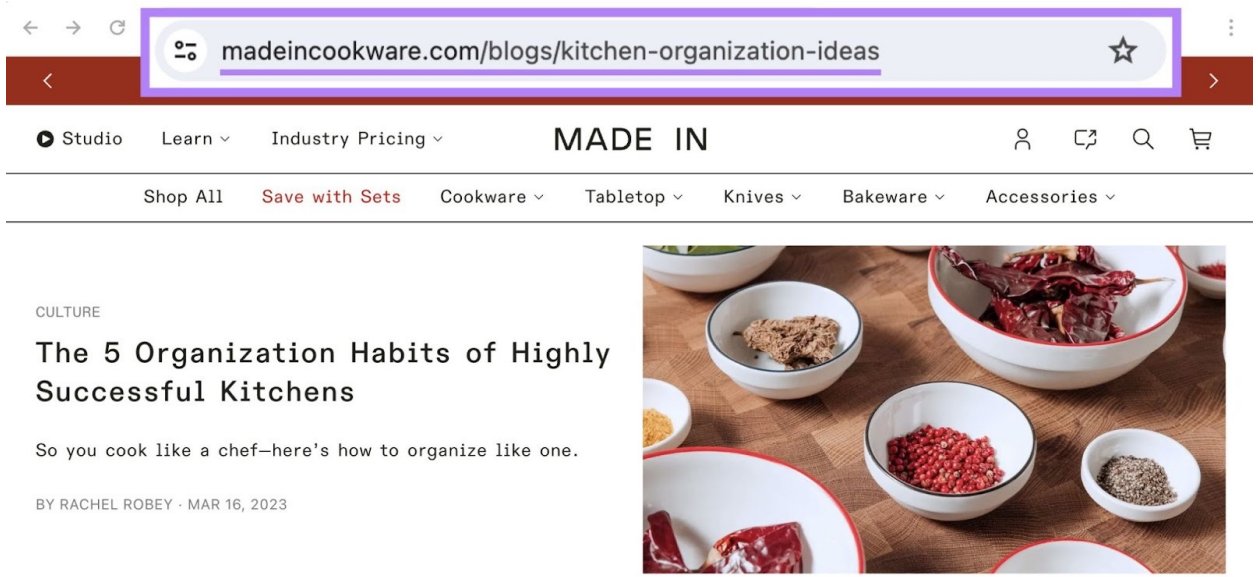
- **Motivation: Pretraining prefers to remove duplicates, ensuring greater coverage with less redundancy**
  - Data Deduplication: Remove duplicates to enhance training or sometimes improve accuracy



  - Utility Function: (1) Exactly match strings or documents or (2) Use approximate matching methods calculated according to similarity measures

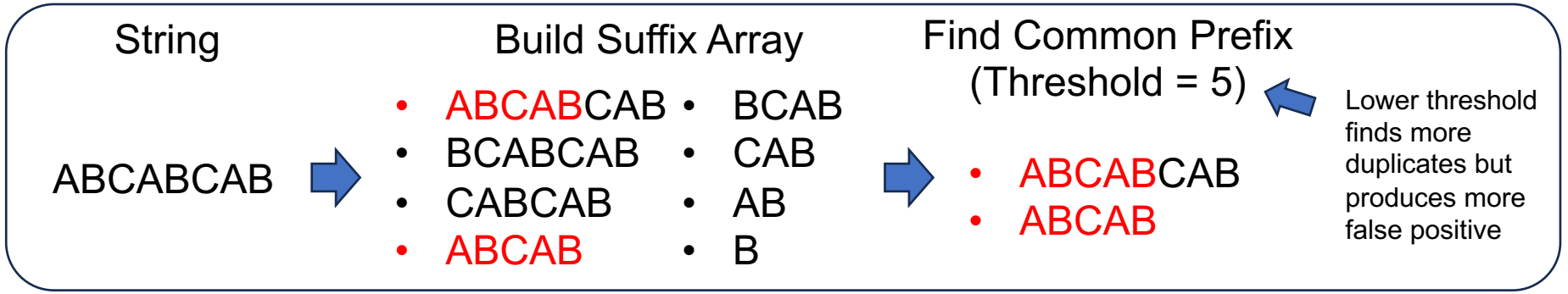
Data Deduplication

- **Motivation: Pretraining prefers to remove duplicates, ensuring greater coverage with less redundancy**
  - Data Deduplication: Remove duplicates to enhance training performance
  - Exact Matching Techniques:
    - 1. URL Deduplication: Remove data that shares the same URL
      - Individual web page may appear in multiple datasets.
      - The samples sourced from the same URL will be identified as duplicates



Data Deduplication

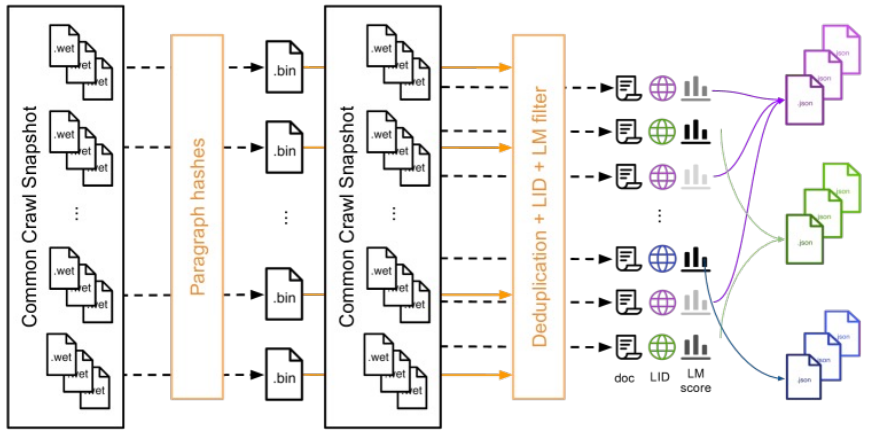
- **Motivation: Pretraining prefers to remove duplicates, ensuring greater coverage with less redundancy**
  - Data Deduplication: Remove duplicates to enhance training performance
  - Exact Matching Techniques:
    - 2. Exact Substring Deduplication: Remove data that shares the same substring
      - Individual content may be referenced by multiple samples.
      - The samples with the same substring will be identified as duplicates.
    - Suffix Array: Find the duplicates through the common prefix of the suffice
      - **S1:** Combine all the samples into one single string
      - **S2:** Build the Suffix Array of the string
      - **S3:** Find duplicates through common prefix



Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., & Carlini, N. (2021). Deduplicating training data makes language models better.

Data Deduplication

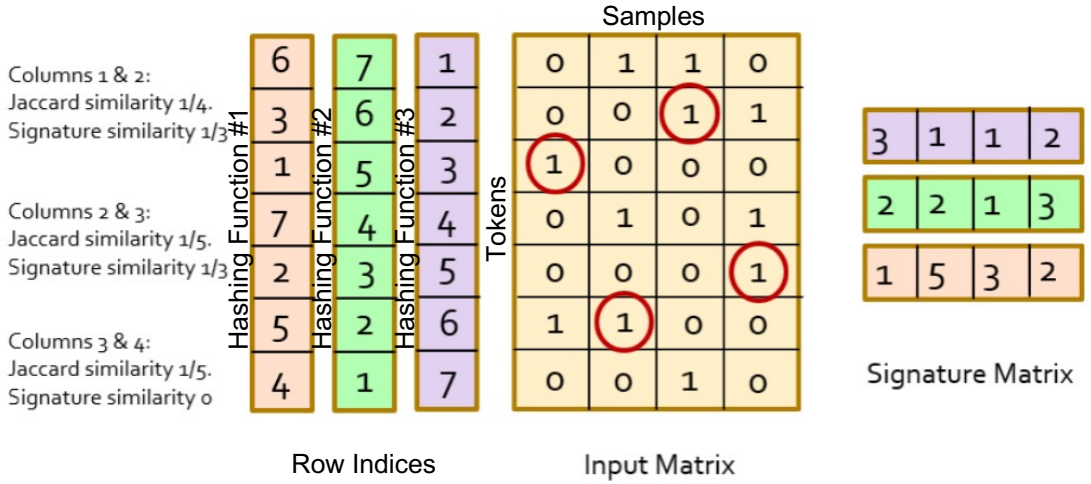
- **Motivation: Pretraining prefers to remove duplicates, ensuring greater coverage with less redundancy**
  - Data Deduplication: Remove duplicates to enhance training performance
  - Approximate Hashing Techniques:
    - 1. Hash Functions: Guarantee to find all exact matches
      - **(1) Initialize a Set for Hashes**
        - A set ~ The hashes of encountered text entries.
      - **(2) Hash Each Text Entry**
        - For each text entry, compute a simple hash (e.g., the sum of ASCII values of its characters).
      - **(3) Check for Duplicates**
        - If the hash of the current entry is already in the set, it is a duplicate and will be ignored.
        - If the hash is not in the set, add the hash to the set and keep the entry.



Efficient and Fast*, but *may find false positives due to hash collisions and remove non-matching documents

Data Deduplication

- **Motivation: Pretraining prefers to remove duplicates, ensuring greater coverage with less redundancy**
  - Data Deduplication: Remove duplicates to enhance training or sometimes improve accuracy
  - Approximate Hashing Techniques (MinHash):
    - **S1:** One-hot encode the samples by, e.g., n-grams, forming a matrix with row as token and column as sample
    - **S2:** Apply a series of hashing functions to row indices to shuffle the rows. For each sample:
      - After applying each hashing function, look for the lowest row index with value 1.
      - The sample signature is a vector of the lowest row indices.
      - Signature also works: Samples with similar set of tokens produce similar signatures.



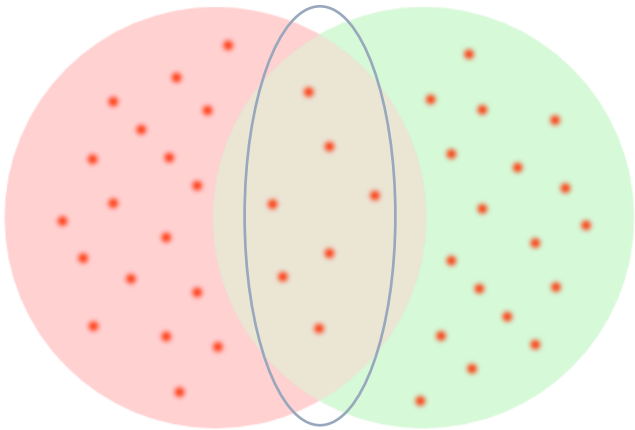
Manku, G. S., Jain, A., & Das Sarma, A. (2007, May). Detecting near-duplicates for web crawling. In Proceedings of the 16th international conference on World Wide Web.

Data Deduplication

- **Motivation: Pretraining prefers to remove duplicates, ensuring greater coverage with less redundancy**
  - Data Deduplication: Remove duplicates to enhance training or sometimes improve accuracy
  - Approximate Hashing Techniques (MinHash):
    - **S1:** One-hot encode the samples by, e.g., n-grams, forming a matrix with row as token and column as sample
    - **S2:** Apply a series of hashing functions to the row indices to obtain random row indices.
    - **S3:** Compute Jaccard Index between fingerprints.

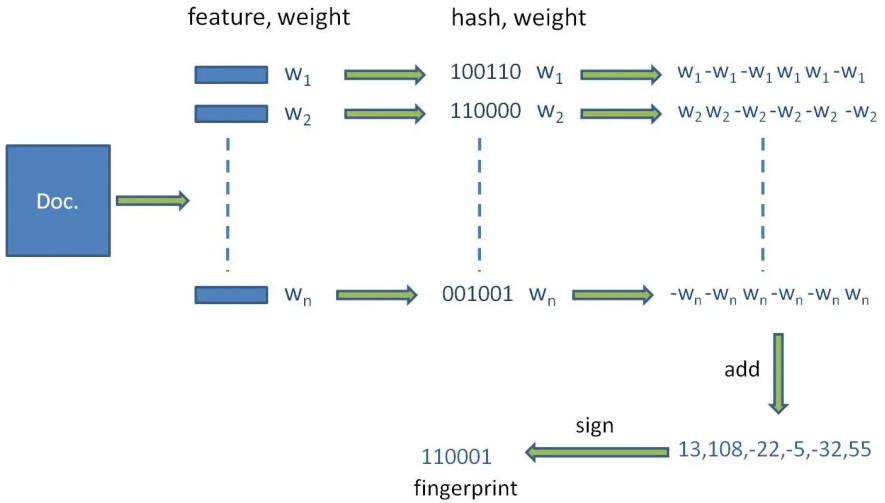
$$\text{Jaccard}(d_i, d_j) = |d_i \cap d_j| / |d_i \cup d_j|$$

      - Less computation by computing Jaccard Index on short fingerprints instead of long original data.
      - The more the hashing functions, the better the signature similarity approximates to the original one



Data Deduplication

- **Motivation: Pretraining prefers to remove duplicates, ensuring greater coverage with less redundancy**
  - Data Deduplication: Remove duplicates to enhance training or sometimes improve accuracy
  - Approximate Hashing Techniques (SimHash):
    - **S1:** Hash each token in the document into a fixed-dimension vector of $\{0, 1\}^d$, weighted by pre-defined or TF-IDF weight w that is positive for 1 and negative for 0.
    - **S2:** Add up weighted vector elements of each token to a single value, forming a new vector of the same dimension d
    - **S3:** Map the new vector to another vector of $\{0, 1\}^d$, which is the fingerprint of each sample
    - **S5:** Compute Hamming distance, the number of different elements between their vectors.



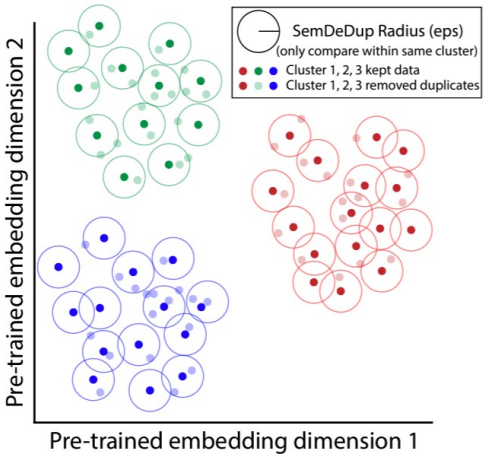
Data Deduplication

- **Motivation: Pretraining prefers to remove duplicates, ensuring greater coverage with less redundancy**
  - Data Deduplication: Remove duplicates to enhance training or sometimes improve accuracy
  - Approximate Frequency Techniques (SoftDeDup): Deduplicate by reweighting samples instead of pruning samples, preventing the loss of potentially valuable information
    - **S1:** Compute the frequency of each n-gram across all the samples
    - **S2:** Calculate the commonness of each sample by multiplying the frequencies of all the n-grams that appear in the document.
    - **S3:** Samples with higher commonness are more likely to be duplicates and thus be down-weighted.

He, N., Xiong, W., Liu, H., et al. SoftDedup: an Efficient Data Reweighting Method for Speeding Up Language Model Pre-training. ACL, 2024.

Data Deduplication

- **Motivation: Pretraining prefers to remove duplicates, ensuring greater coverage with less redundancy**
  - Data Deduplication: Remove duplicates to enhance training or sometimes improve accuracy
  - Embedding-Based Clustering Techniques:
    - Use pretrained models for semantic deduplication
    - **S1:** Embed each data point into a vector in the embedding space using existing LLM. The vector contains the tokens and the context between them, thus providing the semantic information.
    - **S2:** Cluster data points using k-means
    - **S3:** Within each cluster, pairwise cosine similarities between data points are calculated.
    - **S4:** For identified duplicates within a cluster, only the point with the lowest cosine similarity to the cluster centroid is kept, and the others are removed.



Abbas, A., Tirumala, K., Simig, D., Ganguli, S., & Morcos, A. S. (2023). Semdedup: Data-efficient learning at web-scale through semantic deduplication.

Data Deduplication

- **Takeaways**
  - Redundant data negatively impacts LLM performance by reducing generalization ability and increasing overfitting.
  - Deduplication improves training efficiency, prevents memorization of repeated patterns, and mitigates bias.
  - Challenges include efficiently **encoding semantic content** for comparison and **scalability** of deduplication methods for large datasets.
  - Future Directions: 1) Enhancing accuracy in detecting semantically similar but structurally different duplicates; 2) Developing fair deduplication strategies that preserve underrepresented groups.

Data Filtering

Category	Objective	Methods
Sample-level Filtering	Remove low-quality samples	Perplexity Measuring [383], [61], [288], [239], [238] Influence Assessment [254], [168] Clustering [45], [436]
		Model Scoring [411], [264], [345] Mixed Methods [285], [84], [126]
Content-level Filtering	Remove partial-noising samples	Privacy Anonymization [275], [268] Image & Video Filtering [437], [216], [390]

Data Filtering

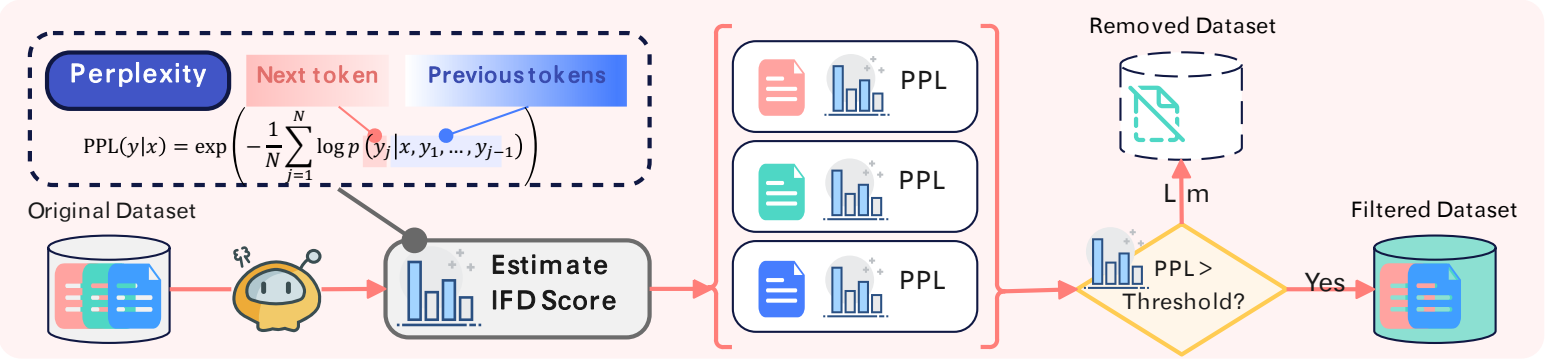
- **Motivation: Pretraining prefers to remove low-quality or noisy samples, simplifying training while retaining performance**
  - Data Filtering: Remove low-quality or noisy samples and ensure diversity in the selected subset.
  - Perplexity Measuring: Filter by sample response generation difficulty.
    - Higher perplexity score indicates higher difficulty for model to generate sample response.
    - Filter samples by generating their responses using an individual model and assessing their perplexity scores.
    - Perplexity score is calculated by conditional probabilities: $\mathrm{PPL}(y|x) = \exp\left(-\frac{1}{N}\sum_{j=1}^{N}\log p\left(y_j|x, y_1, \ldots, y_{j-1}\right)\right)$
    - Given a sentence "I love machine learning", its perplexity score is calculated by

$$P(\mathrm{i}) = 0.2,\ P(\mathrm{love} \mid \mathrm{i}) = 0.1,\ P(\mathrm{machine} \mid \mathrm{i,love}) = 0.05,\ P(\mathrm{learning} \mid \mathrm{i,love,machine}) = 0.01$$

$$PPL(\mathrm{learning|i,love,machine}) = \exp\left(-\frac{1}{4}(\log(0.2) + \log(0.1) + \log(0.05) + \log(0.01))\right) = 17.78279$$

Data Filtering

- **Motivation: Pretraining prefers to remove low-quality or noisy samples, simplifying training while retaining performance**
  - Data Filtering: Remove low-quality or noisy samples and ensure diversity in the selected subset.
  - Perplexity Measuring: Filter by sample response generation difficulty.
    - **S1:** Compute perplexity score with a surrogate model (smaller size for faster computing).
    - **S2:** Rank samples by perplexity values and select subsets based on the criteria (e.g., top-ranked domains or medium/high perplexity samples).
    - **S3:** Train larger models on the optimal subset.



Ankner, Z., Blakeney, C., Sreenivasan, K., Marion, M., Leavitt, M. L., & Paul, M. (2024). Perplexed by perplexity: Perplexity-based data pruning with small reference models.

Data Filtering

- **Motivation: Pretraining prefers to remove low-quality or noisy samples, simplifying training while retaining performance**
  - Data Filtering: Remove low-quality or noisy samples and ensure diversity in the selected subset.
  - Perplexity Measuring: Filter by sample response generation difficulty.
    - **Learning Percentage (LP):** Models learn easier samples first and harder ones later. Assess sample difficulty by measuring its perplexity drop ratio during training.
    - **S1:** Train a model to track sample perplexity across epochs.
    - **S2:** Calculate the learning percentage after the first epoch $LP(1)$.
    - **S3:** Rank samples and split them into three parts (hardest, medium, easiest).
    - **S4:** Train larger models on the hardest subset.

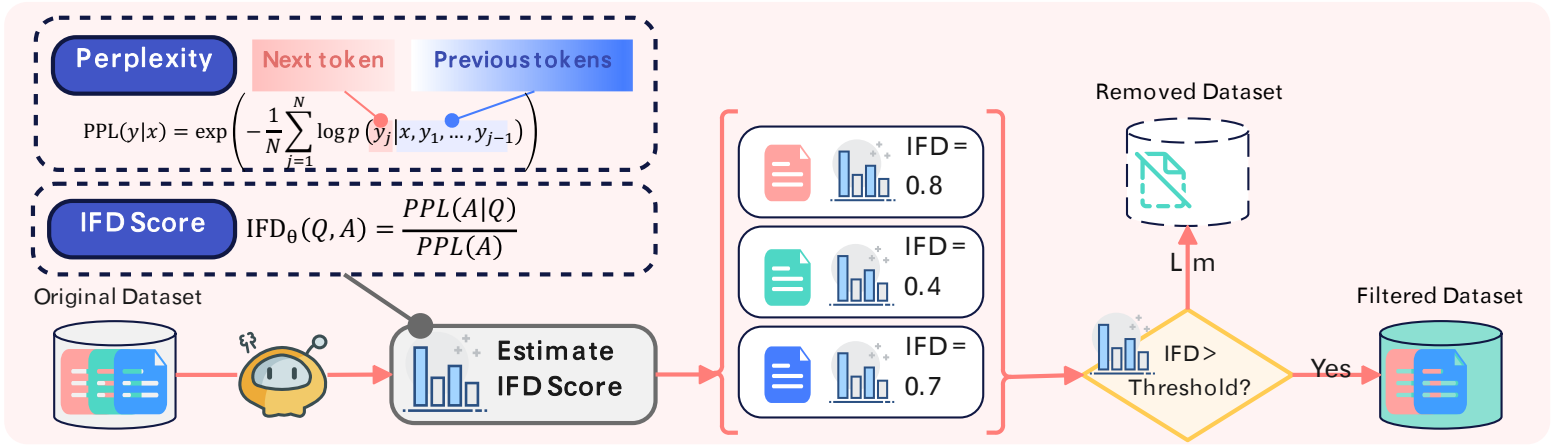
$$\mathcal{LP}(i) = \frac{\mathcal{P}_{i-1} - \mathcal{P}_i}{\mathcal{P}_0 - \mathcal{P}_n}$$

measures the perplexity drop ratio of a sample between the specific epoch i and the whole training procedure.

Mekala, D., Nguyen, A., & Shang, J. (2024). Smaller language models are capable of selecting instruction-tuning training data for larger language models.

Data Filtering

- **Motivation: Pretraining prefers to remove low-quality or noisy samples, simplifying training while retaining performance**
  - Data Filtering: Remove low-quality or noisy samples and ensure diversity in the selected subset.
  - Perplexity Measuring: Filter by sample response generation difficulty.
    - **Instruction-Following Difficulty (IFD):** Measures how much the instruction + input part of the sample would affect sample perplexity by comparing perplexity w/o the part.
    - **S1:** Compute IFD score using an existing model.
    - **S2:** Rank samples by IFD scores.
    - **S3:** Train model on the samples with higher IFD scores.



Li, M., Zhang, Y., Li, Z. et al.: From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning.
Li, M., Zhang, Y., He, S. et al.: Superfiltering: Weak-to-strong data filtering for fast instruction-tuning.

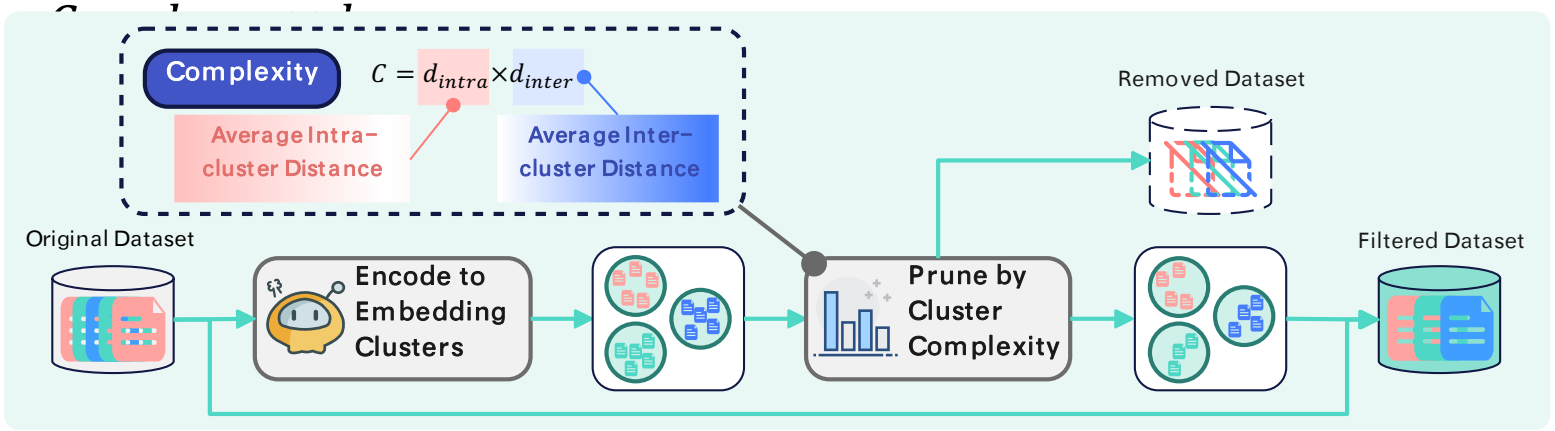
Data Filtering

- **Motivation: Pretraining prefers to remove low-quality or noisy samples, simplifying training while retaining performance**
  - Data Filtering: Remove low-quality or noisy samples and ensure diversity in the selected subset.
  - Influence Assessment: Filter by how much a sample would affect model parameters or model performance (e.g., accuracy, fairness, etc.).
    - **DEALRec:** Estimate the influence of a sample when removing/upweighting samples.
      - Estimate model parameter change through $\widehat{\theta_{-s}} - \hat{\theta} \approx \frac{1}{n} H_{\hat{\theta}}^{-1} \nabla_{\theta \mathcal{L}}(s, \hat{\theta})$, and extend it to
      - Estimate model performance change through $I_{\text{remove, loss}}(s, \mathcal{D}) = \frac{1}{n} \sum_i \frac{1}{n} \nabla_{\theta \mathcal{L}}(s_i, \hat{\theta})^{\top} H_{\hat{\theta}}^{-1} \nabla_{\theta \mathcal{L}}(s, \hat{\theta})$
    - **SHED:** Assess the influence of a sample on model performance using Shapley value.
      - Iteratively removing n samples to measure their contribution to model performance by comparing model performance w/o this subset.

Lin, X., Wang, W., Li, Y., Yang, S., Feng, F., Wei, Y., & Chua, T. S. (2024, July). Data-efficient Fine-tuning for LLM-based Recommendation. In Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval (pp. 365-374).
He, Y., Wang, Z., Shen, Z., Sun, G., Dai, Y., Wu, Y., ... & Li, A. (2024). Shed: Shapley-based automated dataset refinement for instruction fine-tuning. arXiv preprint arXiv:2405.00705.

Data Filtering

- **Motivation: Pretraining prefers to remove low-quality or noisy samples, simplifying training while retaining performance**
  - Data Filtering: Remove low-quality or noisy samples and ensure diversity in the selected subset.
  - Clustering: Groups similar samples together, allowing selection within clusters to reduce redundancy and across clusters to increase diversity.
    - **S1:** Encode samples into embeddings and cluster similar samples using cosine similarity.
    - **S2:** Calculate cluster complexity based on intra-cluster and inter-cluster distances:
    - **S3:** Resample clusters probabilistically to balance diversity and quality.



Abbas, A., Rusak, E., Tirumala, K., Brendel, W., Chaudhuri, K., & Morcos, A. S. (2024). Effective pruning of web-scale datasets based on complexity of concept clusters.

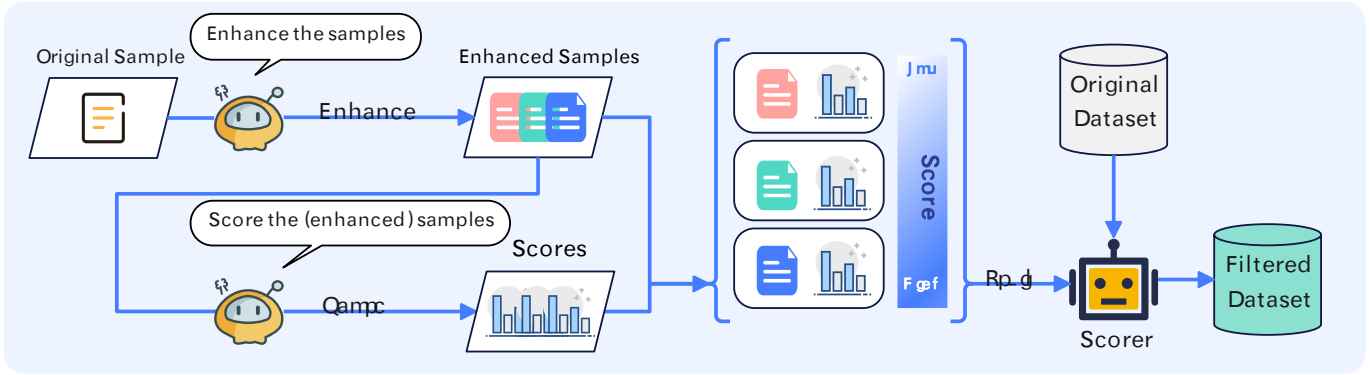
Data Filtering

- **Motivation: Pretraining prefers to remove low-quality or noisy samples, simplifying training while retaining performance**
  - Data Filtering: Remove low-quality or noisy samples and ensure diversity in the selected subset.
  - Model Scoring: Use LLMs to evaluate sample quality explicitly or implicitly through prompt engineering or human-labeled data.
    - **S1:** Prompt GPT-3.5-turbo to compare pairs of samples along four quality criteria (writing style, fact & trivia amount, educational value, and the expertise required to understand)
    - **S2:** Record binary confidence $p_{B \succ A} \in [0,1]$ and translate it into probabilistic sample quality rating $p_{B \succ A} = \sigma(s_B - s_A)$ through Bradley-Terry model.
    - **S3:** Train a rating model on these quality ratings.
    - **S4:** Use the rating model predict quality ratings for new samples on each criterion.
    - **S5:** Resample new samples by the predicted ratings.

Wettig, A., Gupta, A., Malik, S., & Chen, D. (2024). Qurating: Selecting high-quality data for training language models.

Data Filtering

- **Motivation: Pretraining prefers to remove low-quality or noisy samples, simplifying training while retaining performance**
  - Data Filtering: Remove low-quality or noisy samples and ensure diversity in the selected subset.
  - Model Scoring: Use LLMs to evaluate sample quality explicitly or implicitly through prompt engineering or human-labeled data.
    - **S1:** Prompt ChatGPT to evolve the samples along instruction complexity and response quality, and to score these evolved samples
    - **S2:** Train two scorers (complexity and quality) on the evolved samples with their scores.
    - **S3:** Use both scorers to score new samples. Multiply both scores to form the final score
    - **S4:** Resample samples by the final score



Liu, W., Zeng, W., He, K., Jiang, Y., & He, J. (2023). What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning.

Data Filtering

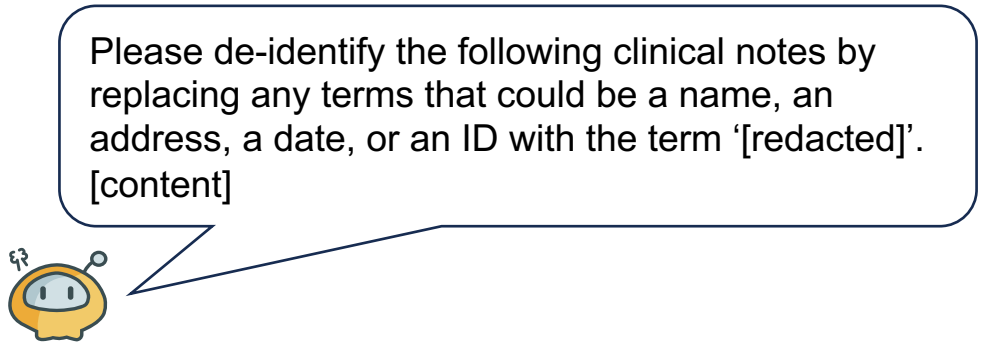
- **Motivation: Pretraining prefers to remove low-quality or noisy samples, simplifying training while retaining performance**
  - Data Filtering: Remove noise or sensitive information within samples.
    - Clean text by removing invalid characters, unnecessary texts, or harmful content (e.g., bias ranging from gender and racial stereotypes to cultural and socioeconomic prejudices).
      - **SEAL:** Train a selector based on a safety-aligned (e.g., Merlinite-7b) model using bi-level optimization:
        - Minimize the safety loss on the safe dataset, and
        - Minimize the fine-tuning loss on the filtered dataset during training

        to make the selector prioritize safe and high-quality samples in selecting data

Navigli, R., Conia, S., & Ross, B. (2023). Biases in large language models: origins, inventory, and discussion. ACM Journal of Data and Information Quality, 15(2), 1-21.
Shen, H., Chen, P. Y., Das, P., & Chen, T. (2024). Seal: Safety-enhanced aligned llm fine-tuning via bilevel data selection.

Data Filtering

- **Motivation: Pretraining prefers to remove low-quality or noisy samples, simplifying training while retaining performance**
  - Data Filtering: Remove noise or sensitive information within samples.
    - Clean text by removing invalid characters, unnecessary texts, or harmful content (e.g., bias).
    - Detect and anonymize private/sensitive information while preserving semantic meaning.
      - Use NER models (spaCy, Flair, etc.) to tag personally identifiable information (e.g., names, addresses) and replace tagged entities with placeholders (e.g., `[NAME]`, `[LOCATION]`), or
      - **DeID-GPT:** Prompt LLM to redact PII within the given text:



Lukas, N., Salem, A., Sim, R., Tople, S., Wutschitz, L., & Zanella-Béguelin, S. (2023, May). Analyzing leakage of personally identifiable information in language models. In 2023 IEEE Symposium on Security and Privacy (SP) (pp. 346-363). IEEE.
Liu, Z., Huang, Y., Yu, X., Zhang, L., Wu, Z., Cao, C., ... & Li, X. (2023). Deid-gpt: Zero-shot medical text de-identification by gpt-4.

Data Filtering

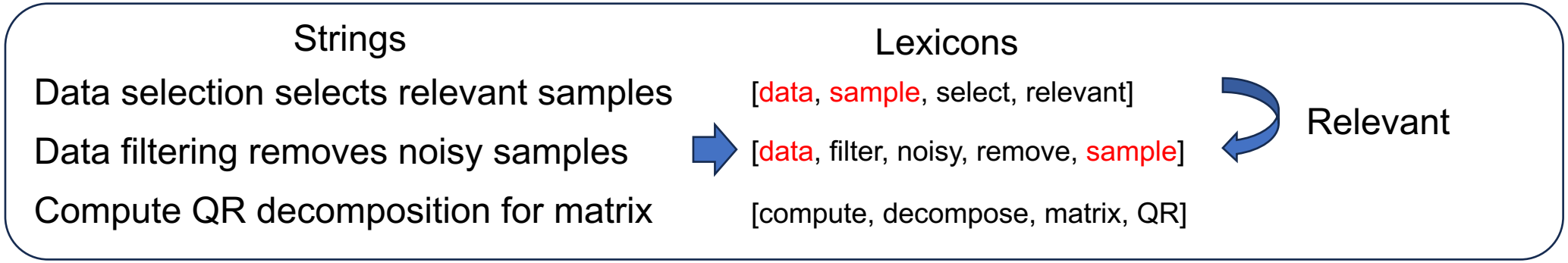
- **Takeaways**
  - Data filtering aims to remove low-quality or sensitive samples from training datasets, reducing computational overhead while maintaining or improving model performance.
  - Challenges include balancing data reduction with model performance and ensuring diversity while filtering out redundant or noisy samples.
  - **Sample-level filtering** focuses on removing entire low-quality samples based on metrics like perplexity, influence assessment, clustering, entropy, and model scoring.
  - **Content-level filtering** targets partial noise or sensitive content within samples rather than removing entire entries.
  - Future Directions include improving efficiency for massive datasets, enhancing accuracy in detecting low-quality or noisy text and developing better algorithms for locating potential sensitive information.

Data Selection

Method	Stage	Evaluation Metric
Similarity	Pre-training, Fine-tuning	Cosine Similarity [423] Bag-of-Words Similarity [421] Lexicon Set Overlap [321] Bayes-based Selection [80]
Optimization	Fine-tuning	Linear Search [130] Gradient-Influence Search [417] Kernel-Density Regularization [269]
Model	Pre-training	Logits-based LM-Score [465]

Data Selection

- **Motivation: Adapt LLMs to specific domains (e.g., medical or legal).**
  - Data Selection: Select subsets of already well-cleaned data samples to align LLMs with target tasks while maintaining generalization.
  - Lexicon Set Overlap: Measures relevance of a dataset to a specific domain using overlap between lexicons.
    - Break strings into lexicon sets, and select samples with large lexicon intersection to the target task

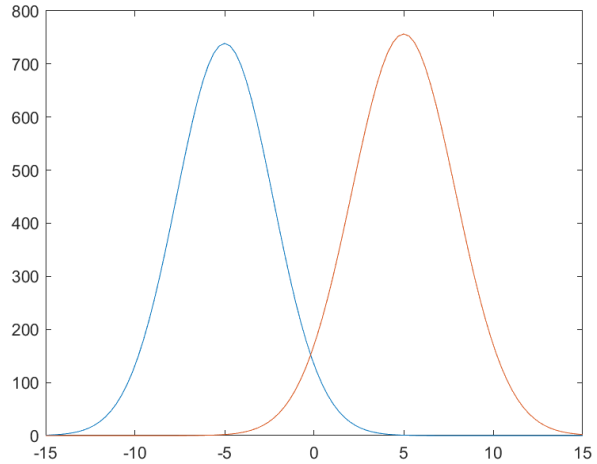


Qin, R., Xia, J., Jia, Z., Jiang, M., Abbasi, A., Zhou, P., ... & Shi, Y. (2024, June). Enabling on-device large language model personalization with self-supervised data selection and synthesis. In Proceedings of the 61st ACM/IEEE Design Automation Conference (pp. 1-6).

Data Selection

- **Motivation: Adapt LLMs to specific domains (e.g., medical or legal).**
  - Data Selection: Select subsets of already well-cleaned data samples to align LLMs with target tasks while maintaining generalization.
  - Bag-of-Words Similarity: Utilize bag-of-words to compute feature distributions for raw and target data.
    - **S1:** Represent raw and target data as bag-of-words features.
    - **S2:** Estimate importance weights $w_i = \frac{\widehat{p_{\text{feat}}}(z_i)}{\widehat{q_{\text{feat}}}(z_i)}$.
    - **S3:** Resample raw data with probability $\frac{w_i}{\sum_{i=1}^{N} w_i}$ to match the target distribution.

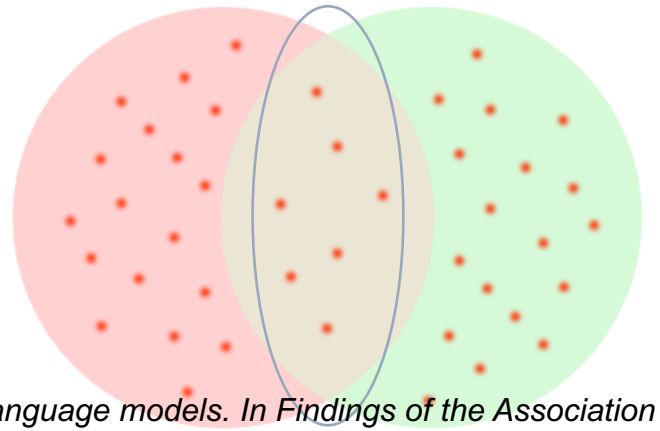
"Data selection selects relevant samples" → p(data) = 1/5, p(sample) = 1/5, p(select) = 2/5, p(relevant) = 1/5 →



Xie, S. M., Santurkar, S., Ma, T., & Liang, P. S. (2023). Data selection for language models via importance resampling. Advances in Neural Information Processing Systems, 36, 34201-34227.

Data Selection

- **Motivation: Adapt LLMs to specific domains (e.g., medical or legal).**
  - Data Selection: Select subsets of already well-cleaned data samples to align LLMs with target tasks while maintaining generalization.
  - Cosine Similarity: Compare embeddings of task-specific labeled data and unlabeled data.
    - **S1:** Encode both labeled and unlabeled data into embeddings
    - **S2:** Measure similarity between embeddings using cosine similarity.
    - **S3:** Select unlabeled samples that align closely with the task's embedding distribution.



Xie, Y., Aggarwal, K., & Ahmad, A. (2024, August). Efficient continual pre-training for building domain specific large language models. In Findings of the Association for Computational Linguistics ACL 2024 (pp. 10184-10201).

Data Selection

- **Motivation: Adapt LLMs to specific domains (e.g., medical or legal).**
  - Data Selection: Select subsets of already well-cleaned data samples to align LLMs with target tasks while maintaining generalization.
  - Optimization-based Selection: Select subsets towards reducing model loss and improving model performance on the target tasks.
    - **Approach 1:** Minimizes model loss on target tasks using linear datamodels.
      - **S1:** Use a linear datamodel $\tau_{\theta_x}(1_S)$ to estimate how training samples affect model loss.
      - **S2:** Adjust characteristic vector 1_S to reflect the subset and estimate parameters θ_x via regression.
      - **S3:** Select the subset S of size k that minimizes the loss $\widehat{L_{D_{\text{targ}}}}(S)$.

Engstrom, L., Feldmann, A., & Madry, A. (2024). Dsdm: Model-aware dataset selection with datamodels.

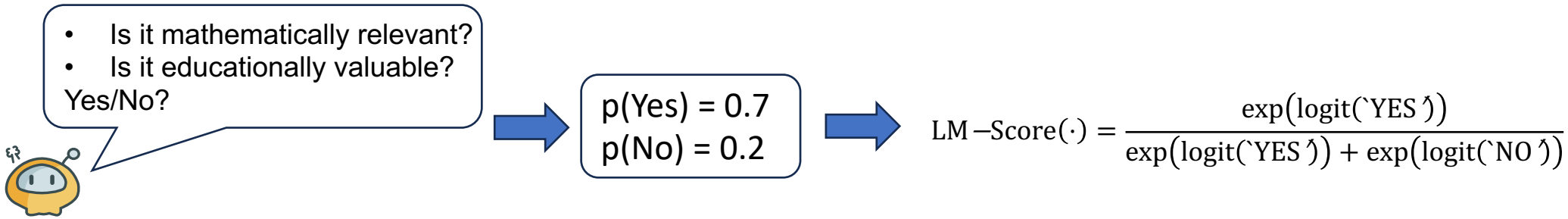
Data Selection

- **Motivation: Adapt LLMs to specific domains (e.g., medical or legal).**
  - Data Selection: Select subsets of already well-cleaned data samples to align LLMs with target tasks while maintaining generalization.
  - Optimization-based Selection: Select subsets towards reducing model loss and improving model performance on the target tasks.
    - **Approach 2:** Identifies impactful data by analyzing gradient similarities.
      - **S1:** Fine-tune the model on a random subset using LoRA.
      - **S2:** Compute Adam LoRA gradients for each sample and project them into lower-dimensional features.
      - **S3:** For downstream tasks, calculate gradient features of validation samples.
      - **S4:** Estimate influence using cosine similarity and select top-scoring samples.

Xia, M., Malladi, S., Gururangan, S., Arora, S., & Chen, D. (2024). Less: Selecting influential data for targeted instruction tuning.

Data Selection

- **Motivation: Adapt LLMs to specific domains (e.g., medical or legal).**
  - Data Selection: Select subsets of already well-cleaned data samples to align LLMs with target tasks while maintaining generalization.
  - Model-Based: Leverages LLMs themselves to evaluate and select high-quality samples.
    - S1: Prompt the LLM to assess relevance and educational value of each sample.
    - S2: Extract logits for responses ("Yes"/"No") and compute LM-Score
    - S3: Calculate composite score and select samples with highest scores.



Zhang, Y., Luo, Y., Yuan, Y., & Yao, A. C. (2024). Autonomous data selection with language models for mathematical texts. In ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models.

Data Selection

- **Takeaways**
  - Data selection involves choosing subsets of already cleaned data to adapt large language models (LLMs) to specific domains, such as medical or legal fields.
  - Three main types of filtering are discussed: Similarity-Based, Optimization-Based and Model-Based selections:
    - Similarity-based methods select samples with similar feature to the target task.
    - Optimization-based methods select samples that improve model performance on the target task.
    - Model-based methods prompt LLM models to select relevant samples to the target task.
  - Challenges include computational efficiency and robust generalization across tasks.
  - Future Directions include improving efficiency for massive datasets, enhancing accuracy in extracting domain patterns and developing better algorithms that generalizes well with the incoming data.

Data Mixing

Taxonomy	Stage	Methods	Traits
Before Training (Human Experience)	Pre-training	Multi-Source Data Adjusting [139], [347]	Intuitive and easy to implement, suitable for rapid experimentation.
		Entropy-Based Mixing [152]	Low computation cost with quality quantification by entropy.
Before Training (Model-Based Optimization)	Pre-training	Linear Regression Model [263]	Only 10% of DoReMi's [420] computational resources are required. Simultaneously train hundreds of small models to accelerate optimization.
	Pre-training	Bivariate Data Mixing Law [152]	Avoid iterative training of proxy models (low computational costs). Show relation between loss and training steps.
	Continual Pre-training	Chinchilla Scaling Law [323]	Support knowledge transferring to new domains (↓ over 95% training costs).
	Pre-training	Exponential Functions [439]	Support datasets without explicit domain division.
	Continual Pre-training	Power-law Function [160]	Compared to single-objective optimization like [323] [160] ensures that domain performance improvement does not compromise general capabilities.
	Pre-training	Classification Model [251]	Reverse engineering for finding the suitable data recipe of LLMs.
During Training (Bilevel Optimization)	Pre-training	Calculate domain contribution by gradient inner products[135]	Requires a proxy model, performances well in OOD datasets.
	Fine-tuning	Dynamically adjust weights by gradient alignment values [302]	Multiple applications like multilingual training, instruction following, large-scale data reweighting
During Training (Distributionally Robust Optimization)	Pre-training	Group DRO [420]	For pre-training, smooth adjusting to prevent abrupt weight changes
	Fine-tuning	Task-level DRO [278]	For fine tuning, quick response to task difficulty changes

Data Mixing For LLM training

- **Challenge: How to optimize dataset ratios for better performance and training efficiency.**
  - Before Training(Human Experience): Empirically set and experiment different ratios of datasets based on various factors (e.g., complexity and diversity of the datasets) that likely improve LLMs' abilities.
    - e.g., Two-phase training, which focuses diversity in data in phase 1, high quality data such as math and code in phase 2, explores their effect with 5 blends each.

Category	Domain	Blend1	Blend2	Blend3	Blend4	Blend5
Web Crawl	-	65.0	65.0	58.0	59.0	70.0
High Quality	Math	1.9	1.9	1.9	2.9	1.9
	Wiki	0.1	0.1	0.1	0.1	0.1
	Code	15.0	8.0	15.0	20.0	13.0
Medium Quality	Books	5.5	9.0	9.0	5.5	4.5
	Papers	3.5	5.0	5.0	3.5	1.9
	CC_{dv}	4.0	6.0	6.0	4.0	3.6
Multilingual	-	5.0	5.0	5.0	5.0	5.0

Table 2: Phase-1 Blends (in %)

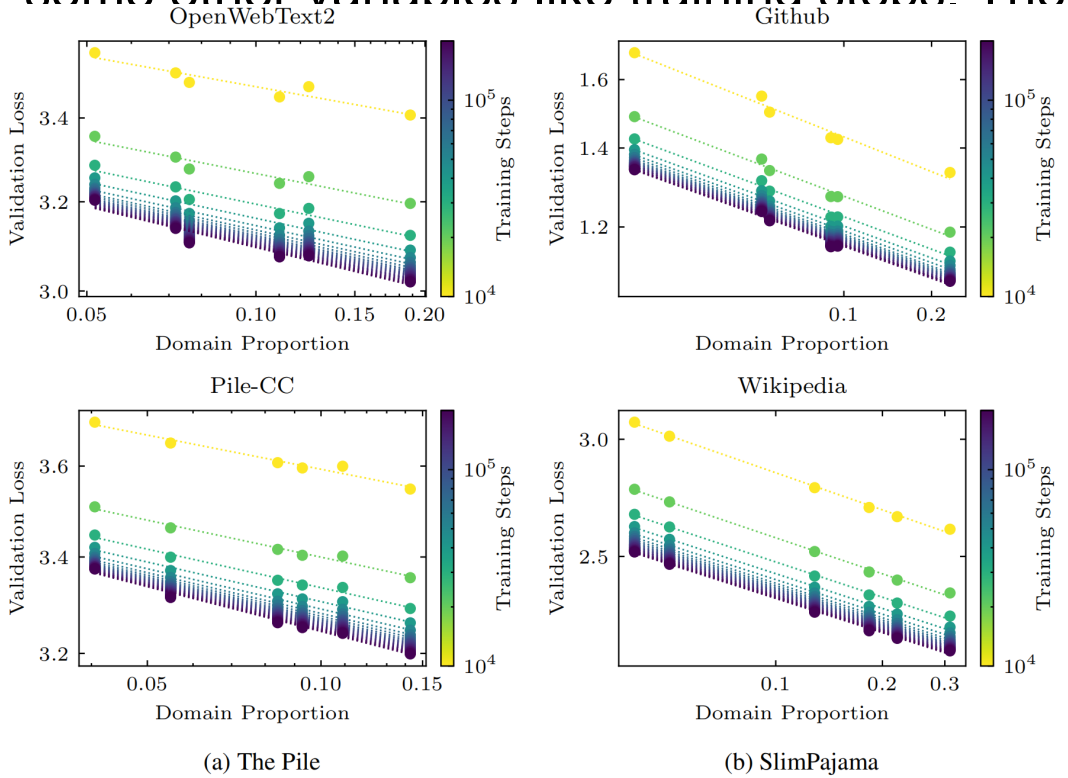
Category	Domain	Blend1	Blend2	Blend3	Blend4	Blend5
Web Crawl	-	31.0	35.0	31.0	40.0	35.0
High Quality	Math	24.0	24.0	24.0	24.0	29.0
	Wiki	1.0	1.0	1.0	1.0	1.0
	Code	20.0	25.0	29.0	20.0	20.0
Medium Quality	Books	8.0	4.0	4.0	4.0	4.0
	Papers	4.0	2.0	2.0	2.0	2.0
	CC_{dv}	7.0	4.0	4.0	4.0	4.0
Multilingual	-	3.7	3.7	3.7	3.7	3.7
Task Data	-	1.3	1.3	1.3	1.3	1.3

Table 3: Phase-2 Blends (in %)

Feng, S., Prabhumoye, S., Kong, K. et al.: Maximize your data's potential: Enhancing llm accuracy with two-phase pretraining. arXiv preprint arXiv:2412.15285 (2024)

Data Mixing For LLM training

- **Challenge: How to optimize dataset ratios for better performance and training efficiency.**
  - Before Training(Model-Based Optimazation): Model the relationship that depicts the relation between (i) the distribution of each domain or datasets, (ii) validation loss, and (iii) some other variables like training steps. Then find the optimal ratio based on the models.



(a) The Pile (b) SlimPajama

e.g., Based on the observation of data from experiments for scaling behaviour, Bivariate Data Mixing Law depicts the relation among domain's proportion, training steps and validation loss.

Ge, C., Ma, Z., Chen, D. et al.: Bimix: A bivariate data mixing law for language model pretraining (2025)

Data Mixing For LLM training

- **Challenge: How to optimize dataset ratios for better performance and training efficiency.**
  - During Training (Bilevel Optimization): Use a closed-loop optimization technique for two nested optimazation problem that ensures model parameters are optimized.
    - e.g., By measuring t**he contribution of a domain to other domains**, which is calculated by the dot product of the gradient of the domain and the sum of gradients from all other domains and the amount of gradients, dynamically adjust the domain by the contributions.

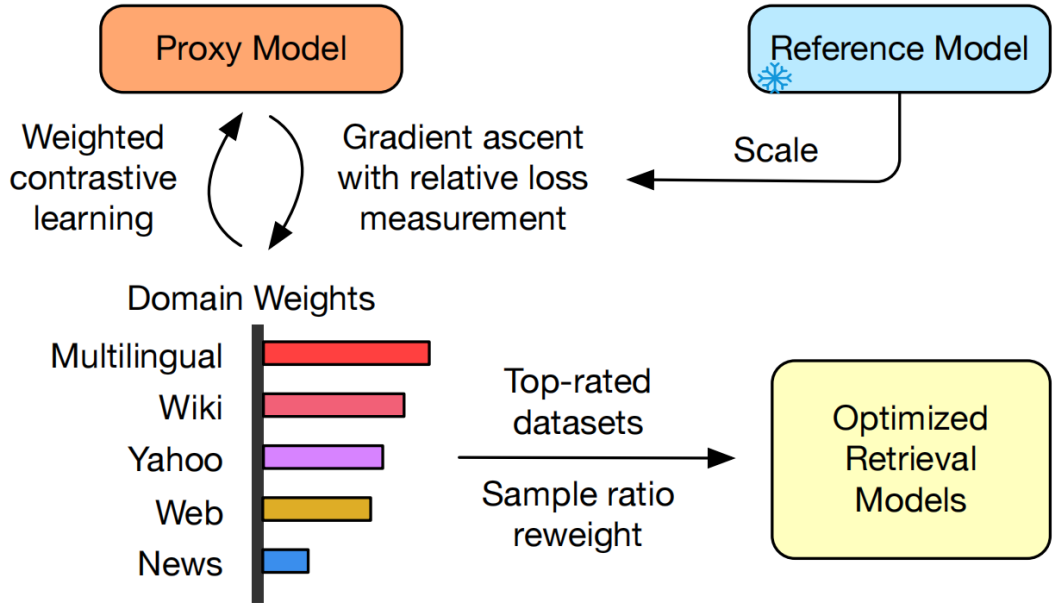
      Minimize the maximum loss across all domains
    - Train a larger model using the optimized domain ratios

Fan, S., Pagliardini, M., Jaggi, M.: Doge: Domain reweighting with generalization estimation. arXiv preprint arXiv:2310.15393 (2023)

Data Mixing For LLM training

- **Challenge: How to optimize dataset ratios for better performance and training efficiency.**
  - During Training (Distributionally Robust Optimization): Adopt Distributionally Robust Optimization (DRO) for a robust data mixing strategy (which can be sub-optimal but with low uncertainty).

e.g., Optimize By DRO using a small proxy model, which first trains a reference model, we have the validation loss of each domain for reference, then trains proxy model, by measuring the **loss difference to reference**, which **indicates the improvement potential**, dynamically adjust the domain weights, and tilt to domains with larger loss difference.



Ma, G., Ma, Y., Wu, X. et al.: Task-level distributionally robust optimization for large language model-based dense retrieval. arXiv preprint arXiv:2408.10613(2024)

Takeaways

- Human experience mixing takes least amount of cost to get a better data ratio for training by using the ratios given by these works, by experimenting like this still needs quite amount of training.
- For now, almost all mixing methods need a small proxy model for experiments then scale up to larger one.
- Model-based methods provide other variables like training step that help us see how these variables affect LLM performance with domain ratios.
- Optimize methods like Bilevel Optimization and DRO performs very well but find difficulties for applying on larger models (Largest model with 34B from recent works), as it’s done during training.

Data Distillation and Synthesis

Stage	Category	Methods
Distillation	Reasoning Augmentation	Cot [353]
		Prompt with Tools [496]
	Data Augmentation	Prompt with Multi-Agent [467]
Pre-Training	Data Augmentation	Distillation + Fine Tuning + Prompt [481]
		Prompt [99], [98], [282], [93],[344], [92]
SFT	Data Augmentation	Prompt [233], [179], [260], [290]
	Reasoning Augmentation	Prompt [178], [173], [346]
		Human Label [253]
		Automated Label [399]
		High Quality Reasoning Data [442], [230]
RL	Prompts Optimization	Prompt [401]
	Human Feedback	RLHF [71]
		RLHF By LLM [476]
RAG	Privacy Protection	Prompt [450]

Data Distillation and Synthesis For LLM training

- **Challenge: How to synthesis better data for tackling various training needs like data scarcity?**
  - Distillation: Design paradigms to prompt LLM to generate high-quality data to train a student LLM with less parameters to mimic the target model's generation ability.

e.g., employs multiple task-specific prompts: (1) Chain-of-Density (CoD): Iteratively adds entities to summarize for enhanced density. (2) Chain-of-Thought (CoT): Guides reasoning tasks (e.g., math) through stepwise logic.

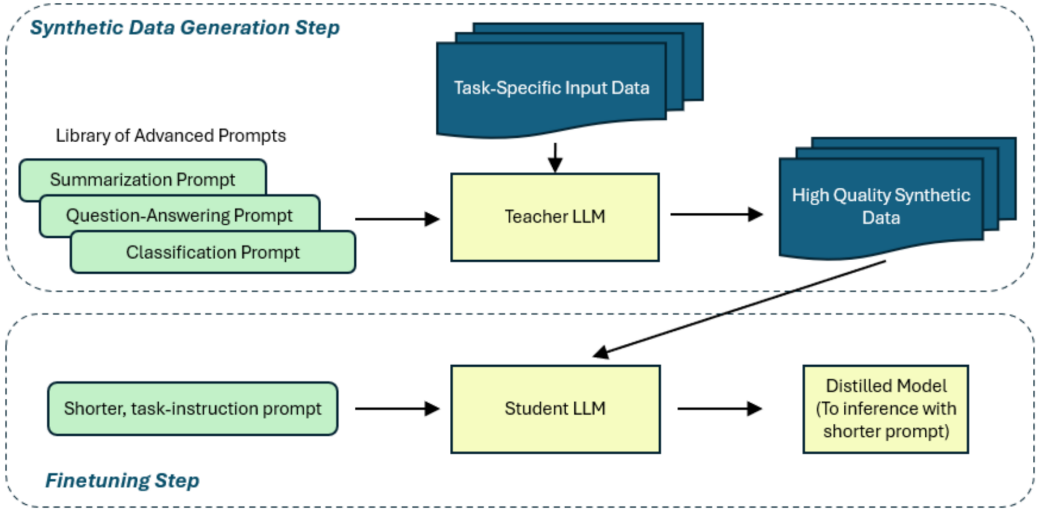


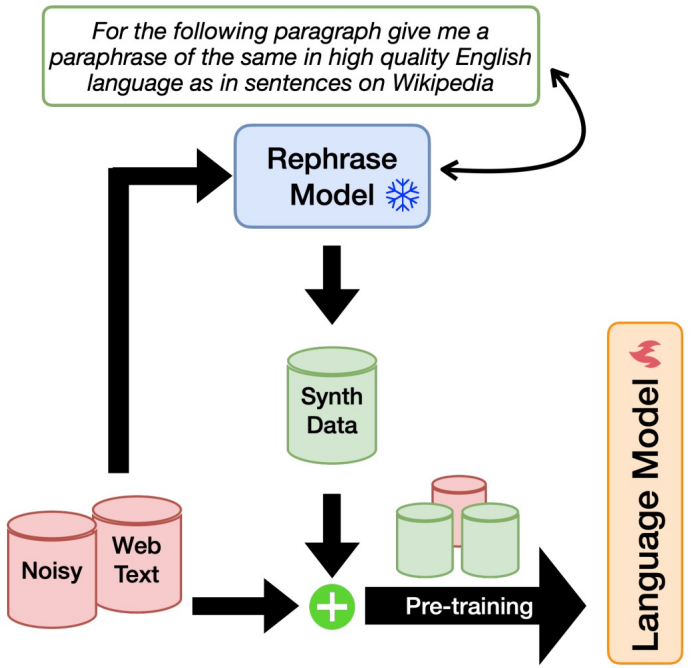
Figure 1: Overview of the proposed methodology for model distillation. Synthetic data is generated using advanced, task-engineered prompt while fine-tuning (hence inferencing) of student model uses shorter, less expensive vanilla prompt.

Shirgaonkar, A., Pandey, N., Abay, N.C. et al.: Knowledge distillation using frontier open-source llms: Generalizability and the role of synthetic data. arXiv preprint arXiv:2410.18588 (2024)

Data Distillation and Synthesis For LLM training

- **Challenge: How to synthesis better data for tackling various training needs like data scarcity?**
  - Pretraining Data Augmentation: Design paradigms to prompt LLM to generate high-quality data and mix them in pretraining.

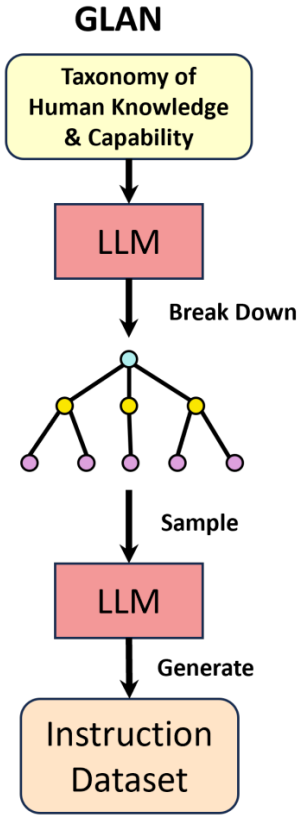
e.g., Leverages instruction-tuned models to rephrase web text into four formats:
(i) simple vocabulary and sentence structures that are understandable to young children.
(ii)Standardized encyclopedia-style expression.
(iii) Complex terminology and concise academic sentence structures.
(iv) multi-turn dialogue. Mixing rephrased and original data trains LLMs to adapt to diverse formats



Maini, P., Seto, S., Bai, H. et al.: Rephrasing the web: A recipe for compute and data-efficient language modeling. arXiv preprint arXiv:2401.16380 (2024)

Data Distillation and Synthesis For LLM training

- **Challenge: How to synthesis better data for tackling various training needs like data scarcity?**
  - SFT Data Augmentation: Design paradigms to prompt LLM to generate high quality data to align LLM's knowledge to instructions, enhancing reasoning ability, etc. focus on improvement of specific domains.

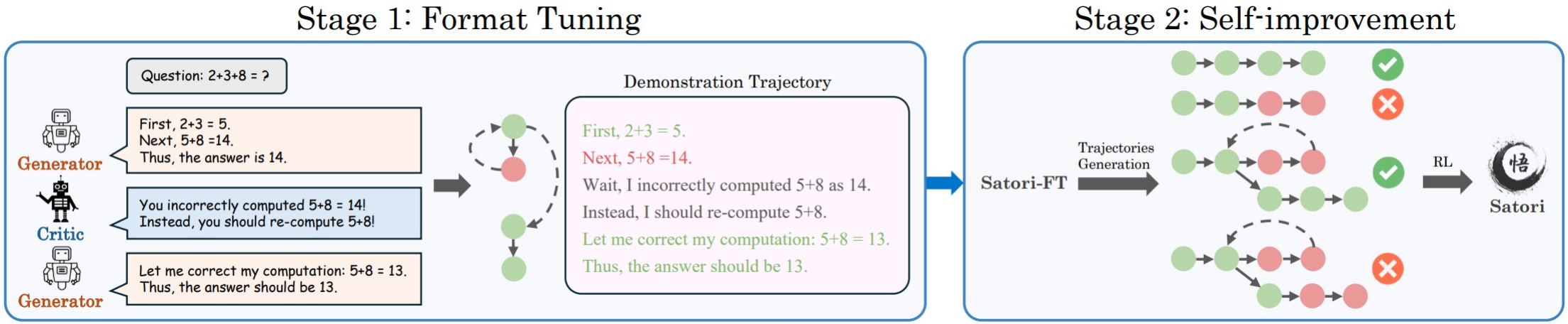


e.g., Introduces a knowledge-classification framework for synthetic text generation by GPT-4. (i) Organize knowledge domains (natural sciences/humanities) into disciplines (math/programming) by; (ii) Develop course outlines with units (e.g., "Intro to Calculus") and core concepts (e.g., "Limits"); (iii) Use GPT-4 to create diverse questions by combining concepts, then generate answers with faster GPT-3.5.

Li, H., Dong, Q., Tang, Z. et al.: Synthetic data (almost) from scratch: Generalized instruction tuning for language models. arXiv preprint arXiv:2402.13064 (2024)

Data Distillation and Synthesis For LLM training

- **Challenge: How to synthesis better data for tackling various training needs like data scarcity?**
  - SFT Reasoning Data Augmentation: Synthesize reasoning data (e.g., code, chain of thought through techniques like Chain-of-thought(CoT), or utilizing verification tools for more rigorous reasoning.
    - e.g., Satori introduce reasoning formats to LLM by fine tuning and RL.



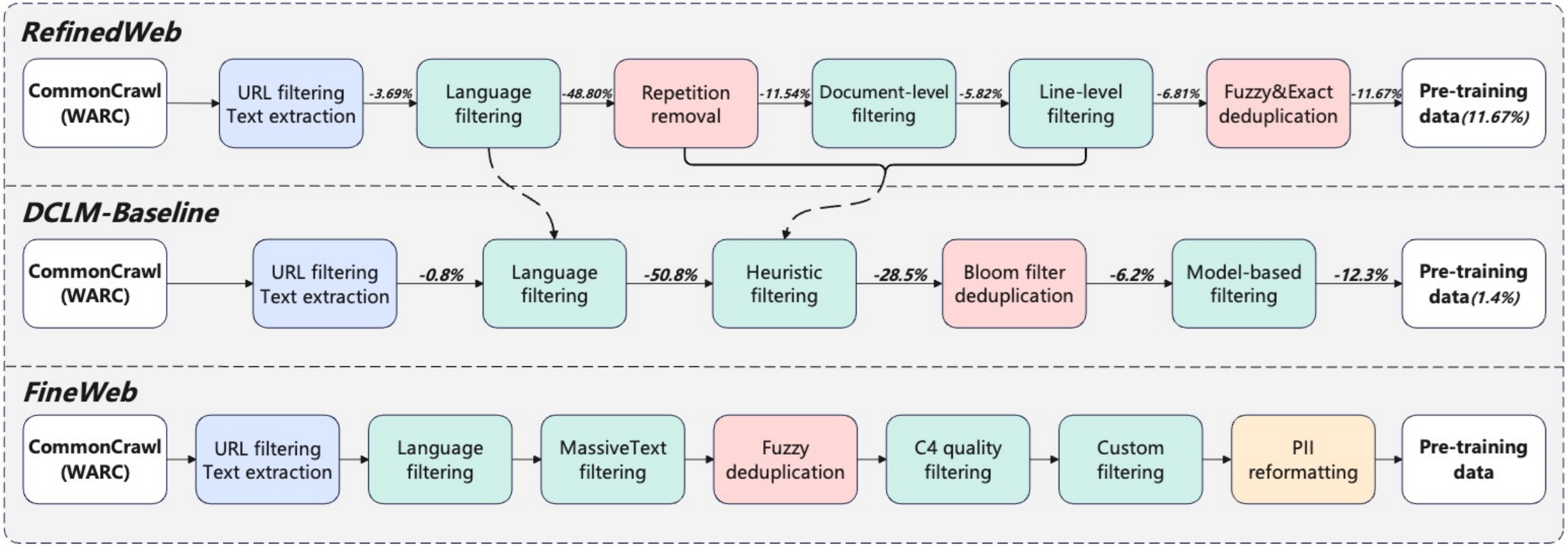
Shen, M., Zeng, G., Qi, Z. et al.: Satori: Reinforcement learning with chain-of-action-thought enhances llm reasoning via autoregressive search. arXiv preprint arXiv:2502.02508 (2025)

Takeaways

- Distillation enables LLMs with less parameters has similar performance of larger LLM.
- Synthetic data for pretraining has to be controlled under certain ratio, should be mixed with authentic data, otherwise it will even does harm to performance.
- High quality and well-formatted reasoning data are keys to high reasoning performance.

End-to-End Pipelines

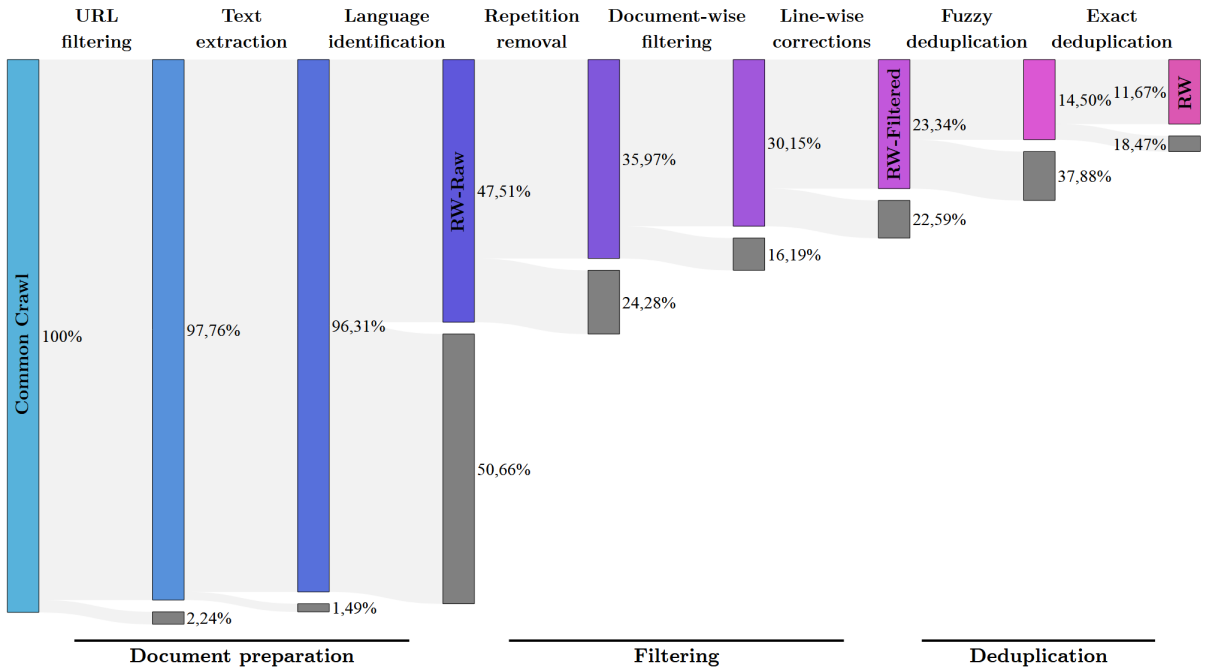
- End-to-End Pipelines: Orchestrate data processing operations that transform **raw data** into **high-quality LLM pre-training data**.



End-to-End Pipelines

- **RefinedWeb Pipeline**

1. **Data acquisition**
   - URL filtering: blocklist & URL score
   - Lang identification: FastText (like Word2Vec)
   - Text extraction: Regx Lib (Trafilatura)
2. **Data filtering**
   - Document-level filtering: Rule-based.
   - Line-level filtering: Rule-based.
3. **Data deduplication**
   - Fuzzy deduplication: Minhash.
   - Exact deduplication: Suffix array.



The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data only[J]. arXiv, 2023.

End-to-End Pipelines

- **DCLM-Baseline Pipeline**
  - Adopt RefinedWeb's heuristic filtering.
  - Use Bloom filter deduplication, offering comparable performance to MinHash with higher efficiency on large-scale datasets.
  - Compared to RefinedWeb, additionally apply model-based filtering, retaining only **1.4%** of raw data (vs. **11.67%** in RefinedWeb).

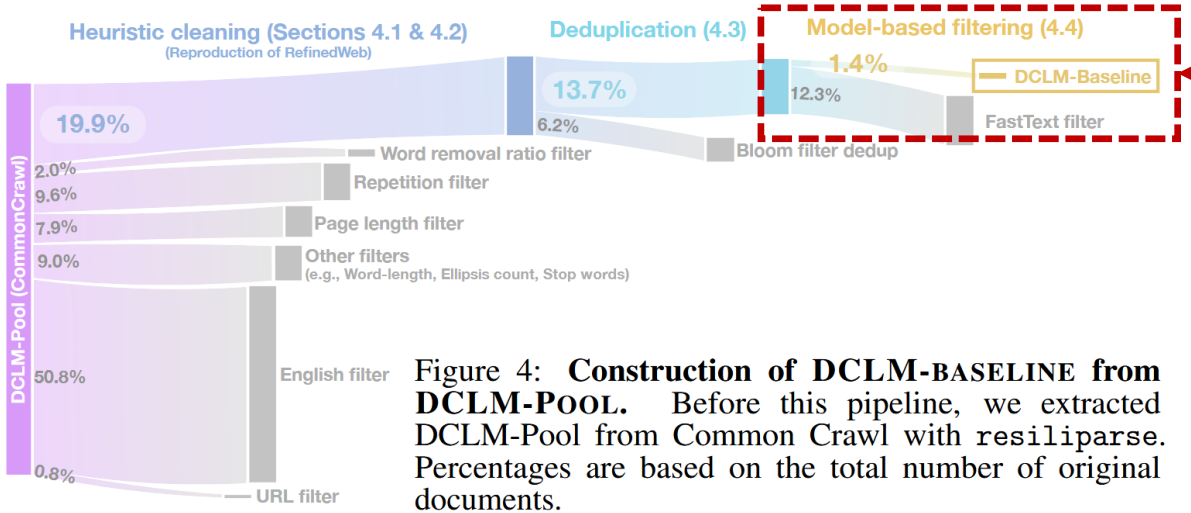


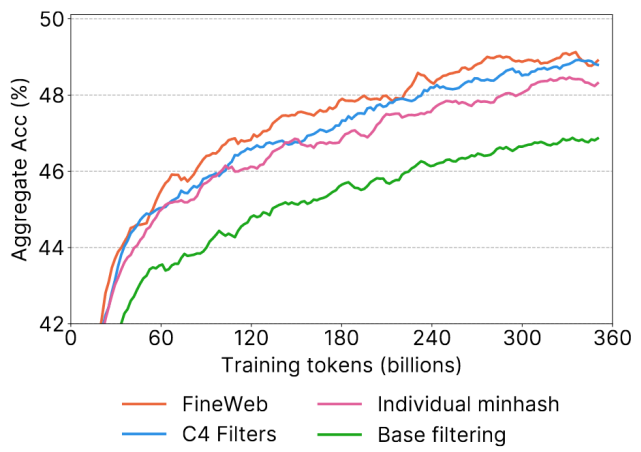
Figure 4: **Construction of DCLM-BASELINE from DCLM-POOL.** Before this pipeline, we extracted DCLM-Pool from Common Crawl with `resiliparse`. Percentages are based on the total number of original documents.

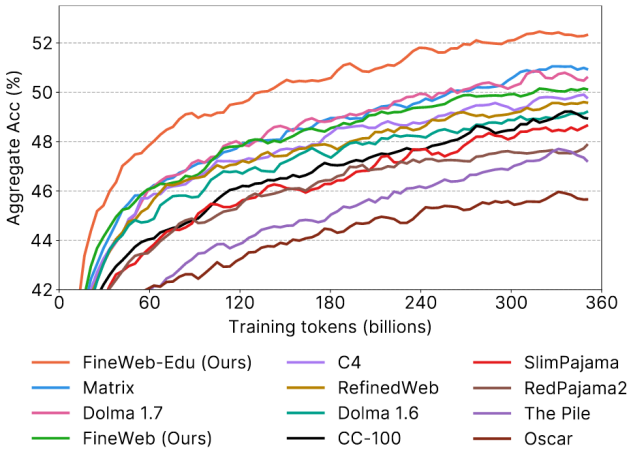
A fastText classifier trained on instruction-formatted data, including diverse data formats (OpenHermes 2.5) and QA samples (ExplainLikeImFive).

Datacomp-lm: In search of the next generation of training sets for language models[J]. NIPS, 2024.

End-to-End Pipelines

- **FineWeb Pipeline**
  - Adopts heuristic filtering from MassiveText and C4 Dataset.
  - Conducts individual Minhash deduplication for each CommonCrawl snapshot.
  - Develops additional custom heuristic filters (e.g. fraction of lines ending with punctuation) through a systematic process for better performance.
  - PII (email addresses and public IP addresses) is anonymized using regex patterns.

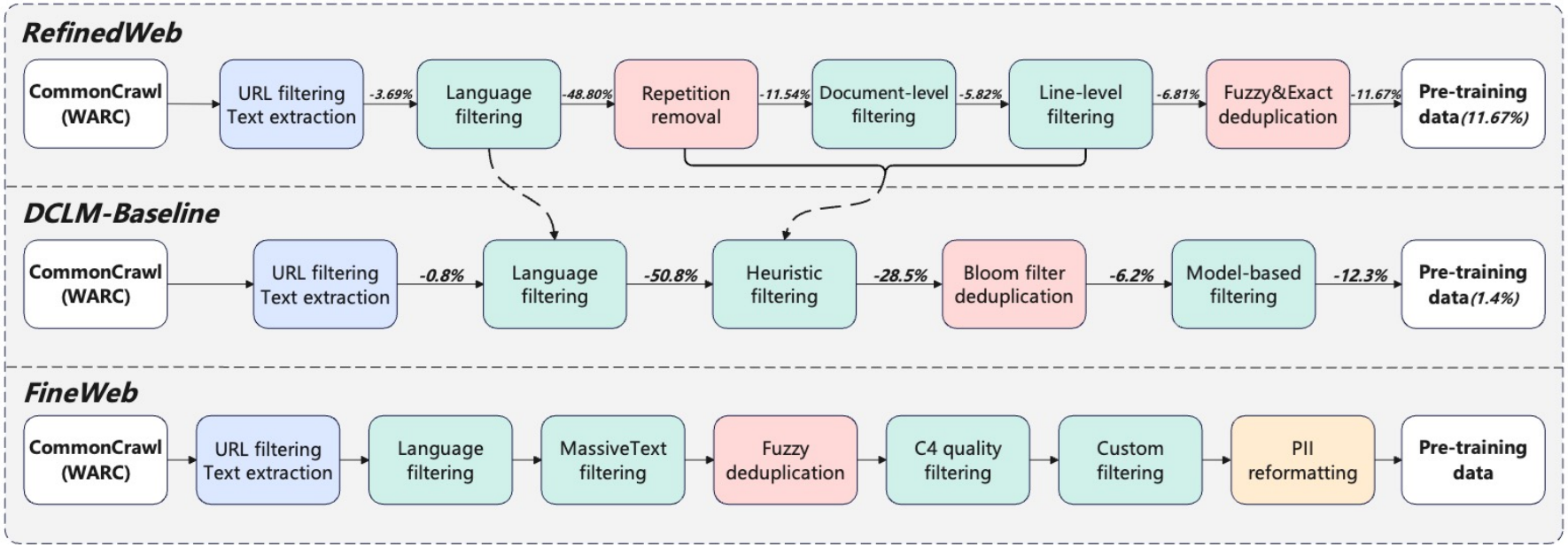




The fineweb datasets: Decanting the web for the finest text data at scale[J]. NIPS, 2024.

End-to-End Pipelines

- **Designing Principles**
  - The trade-off between data quality and quantity.
  - Dependencies across the processing operations (e.g., text extraction necessarily preceding operations like deduplication and filtering).
  - Efficiency optimization (e.g., conducting computationally intensive steps like model-based filtering after lightweight processing steps like URL filtering).



Data Packing For Data-Centric Training

- **Challenge: How to conduct data-centric training on the basis of a high-quality dataset.**

Data Packing Is Required During Pre-Training: Data Packing combines texts to ensure uniform input lengths in pre-training, improving coherence and reducing padding and truncation.

e.g., Best-fit Packing

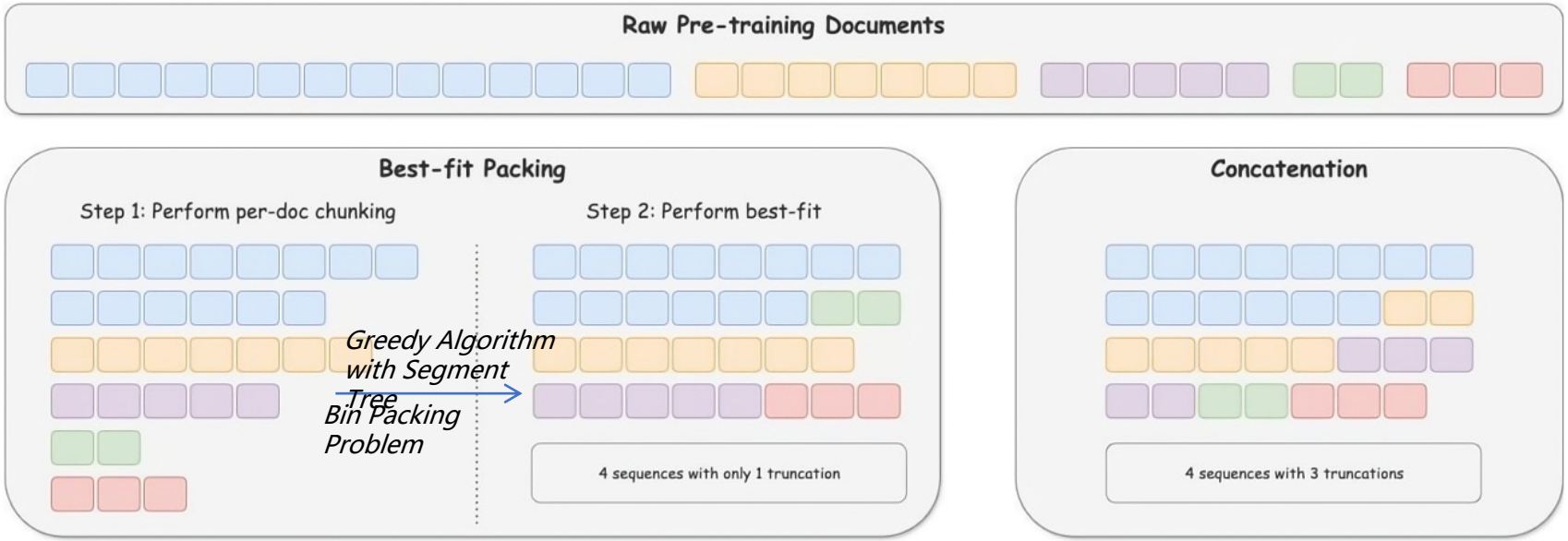


Figure 1. An illustration of the proposed Best-fit Packing compared with concatenation (baseline). We set max sequence length to 8 tokens in this example. **Top:** Original training documents. Each box stands for a token. Contiguous boxes in the same color represent a document. There are five documents of lengths 14, 7, 5, 2, 3, respectively. **Bottom-left:** Best-fit Packing. In step 1, we segment the long document (e.g., blue) into chunks with ≤ 8 tokens. In step 2, we group chunks into training sequences in a smart way that results in the smallest number of sequences. We do not break any chunk in the second step. In total, only one document was truncated and this is necessary to meet the max sequence length requirement. **Bottom-right:** The concatenation approach. 3 out of the 5 documents are truncated.

Hantian Ding, Zijian Wang, et al. Fewer Truncations Improve Language Modeling. ICML, 2024

Training Strategy For Data-Centric Training

- **Challenge: How to conduct data-centric training on the basis of a high-quality dataset.**

During Training Need Appropriate Strategy: Different training strategies lead to different training results.

e.g., Dual-stage Mixed Fine-tuning (DMT) Strategy

First fine-tunes on specific datasets.
Second fine-tunes on mixed data.
To balance general and specialized abilities.

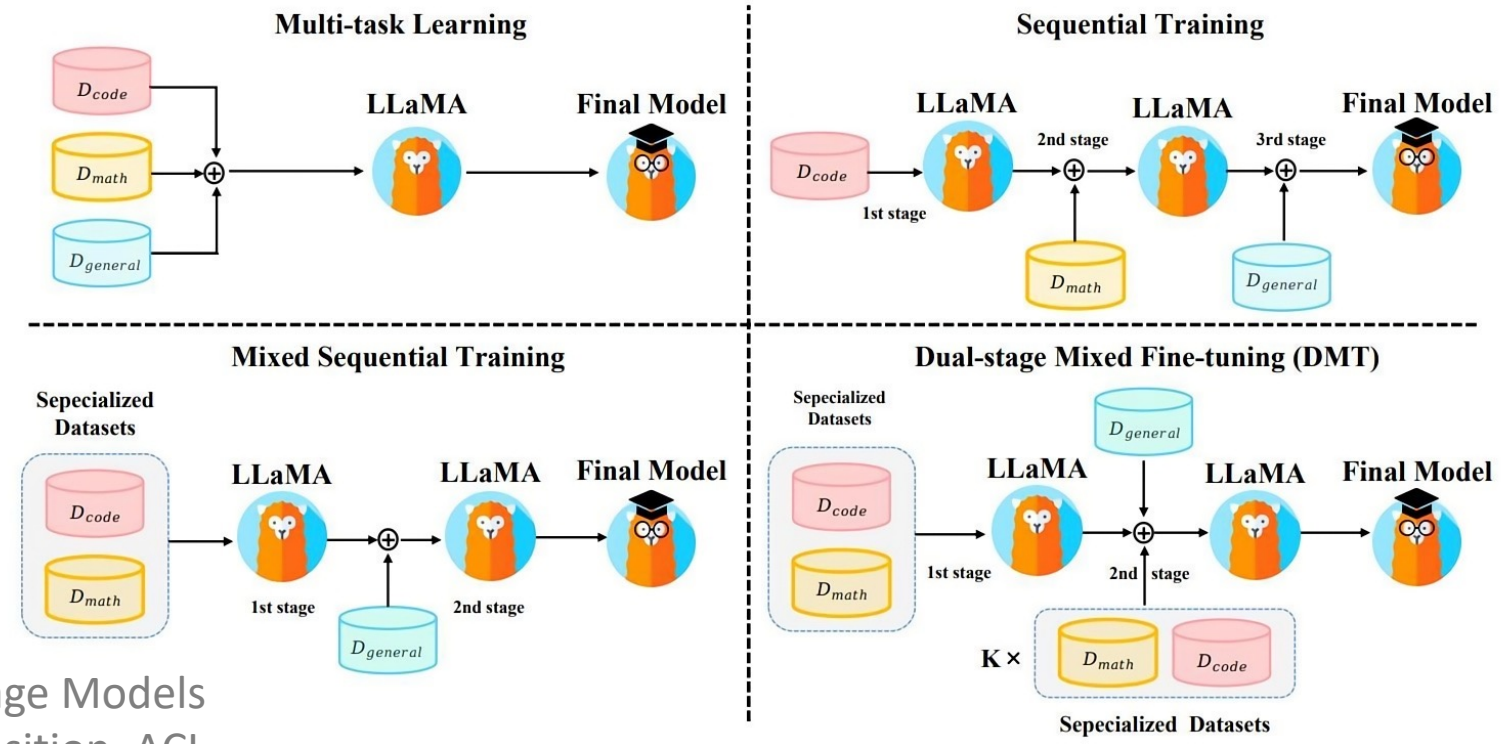


Figure 1: The illustration of four different training strategies in this paper.

Guanting Dong, et al. How Abilities in Large Language Models are Affected by Supervised Fine-tuning Data Composition. ACL, 2024

Training Strategy For Data-Centric Training

- **Challenge: How to conduct data-centric training on the basis of a high-quality dataset.**

Methods	LLaMA -7B			LLaMA -13B			LLaMA -33B		
	GSM8K	HumanEval	MT-Bench	GSM8K	HumanEval	MT-Bench	GSM8K	HumanEval	MT-Bench
Individual domain									
General only	11.10	10.42	5.88	14.02	16.40	6.13	26.06	24.30	6.63
Math only	49.10	6.71	2.53	51.40	12.8	2.54	57.91	15.5	3.18
Code only	4.51	18.40	4.30	5.15	17.1	3.53	6.06	26.82	4.18
Different Training Strategies									
Multi-task learning	**47.53**	14.63	5.76	**50.94**	<u>19.50</u>	5.73	**56.69**	18.9	6.07
Sequential Training	31.39	<u>15.85</u>	5.72	39.12	**20.12**	<u>5.93</u>	47.27	<u>24.80</u>	**6.73**
Mixed Sequential Training	32.60	15.24	<u>6.02</u>	40.48	18.30	<u>5.93</u>	44.24	24.4	6.43
DMT(k=1/256)	<u>41.92</u>	**17.68**	**6.08**	<u>46.47</u>	<u>19.50</u>	**6.03**	<u>56.36</u>	**25.00**	**6.73**

Table 1: The results of LLaMA-7B, 13B, 33B under different training strategies on three benchmarks. The top two results across different strategies are marked with **bold** and <u>underlined</u>.

Figure 1: The illustration of four different training strategies in this paper.

Data Shuffling For Data-Centric Training

- **Challenge: How to conduct data-centric training on the basis of a high-quality dataset.**

Data Shuffling To Help Training: Data shuffling means that different data needs to be selected and provided to LLMs at various stages. (e.g., in different epochs for SFT).

e.g., Velocitune

$$V_t[i] = \frac{\ell_t[i] - \ell_{\text{target}[i]}}{\ell_{\text{init}[i]} - \ell_{\text{target}[i]}}$$

$V_t[i]$ is the learning velocity for domain i at step t

$\ell_t[i]$ is the current loss for domain i

$\ell_{\text{target}[i]}$ is the target loss for domain i, predicted by the scaling law

$\ell_{\text{init}[i]}$ is the initial loss for domain i, calculated before training starts.

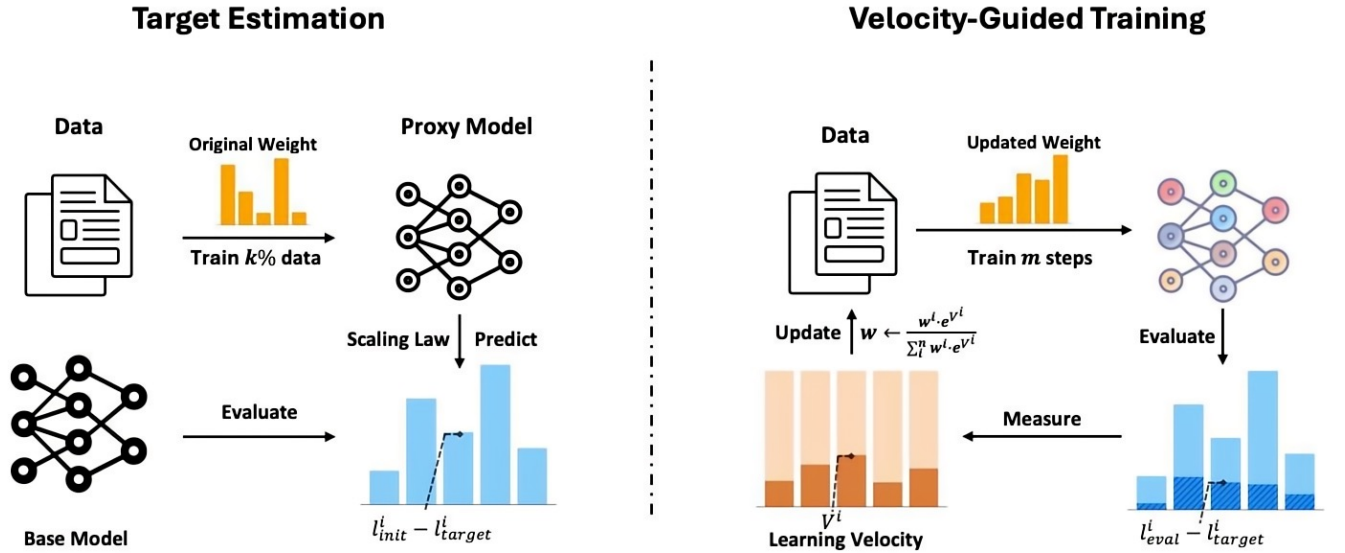


Figure 1: The overall pipeline of Velocitune. Initially, a proxy model is trained using the original domain weights on a subset of the data. Following this, the initial loss is collected by evaluating the base model, while the target loss is determined by extrapolating the evaluation loss of the proxy model. In the second phase, we calculate the learning velocity by rescaling the learning progress between the initial and target losses. This learning velocity is then used to update the domain weights effectively.

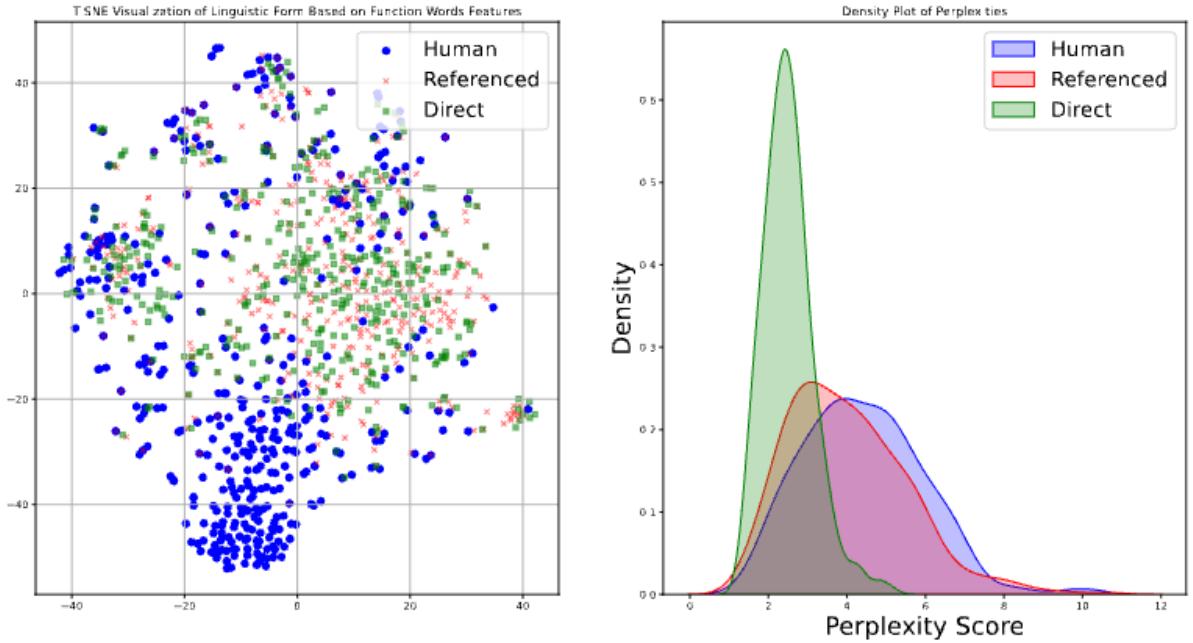
Zheheng Luo, Xin Zhang,et al. Velocitune: A Velocity-based Dynamic Domain Reweighting Method for Continual Pre-training. ACL, 2025

Takeaways

- Data Packing is essentially a bin packing problem in the field of optimization, necessitating the use of efficient algorithms with low time complexity.
- Training Strategy requires an appropriate workflow that explores suitable domain compositions and mixing ratios at each stage.
- Data Shuffling involves monitoring training signals such as loss, gradients... allowing for dynamic adjustment of data sampling ratios throughout the training process.
- Open problems:
  - More explainable Training Strategy
  - More unified metrics for monitoring training status in Data Shuffling

Future Opportunities

- **Task-Specific Data Selection for Efficient Pretraining**
  - Inclusion of irrelevant data not only increases training time but also impedes the model's adaptability to specific tasks → Adaptive data selection strategies

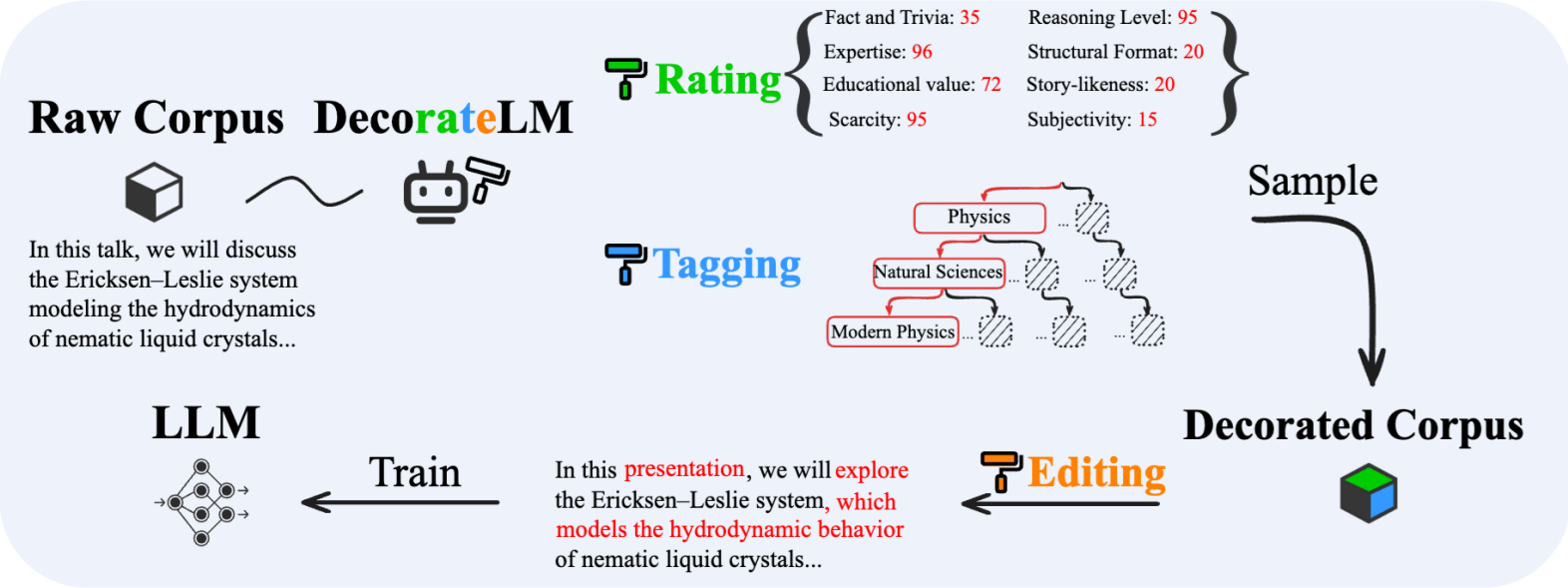


Align with Human Preference → Specific Tasks

SCAR: Data Selection via Style Consistency-Aware Response Ranking for Efficient Instruction-Tuning of Large Language Models. arXiv, 2024.

Future Opportunities

- **Predictive Pipeline Selection / Processing Agent Design**
  - Experimentally decide the pipeline is resource-intensive → Predict optimal preprocessing configurations in advance or design agentic processing method



Empirical Pipelines → Data(-agent) driven Pipelines

DecorateLM: Data Engineering through Corpus Rating, Tagging, and Editing with Language Models. EMNLP, 2024.

Thanks