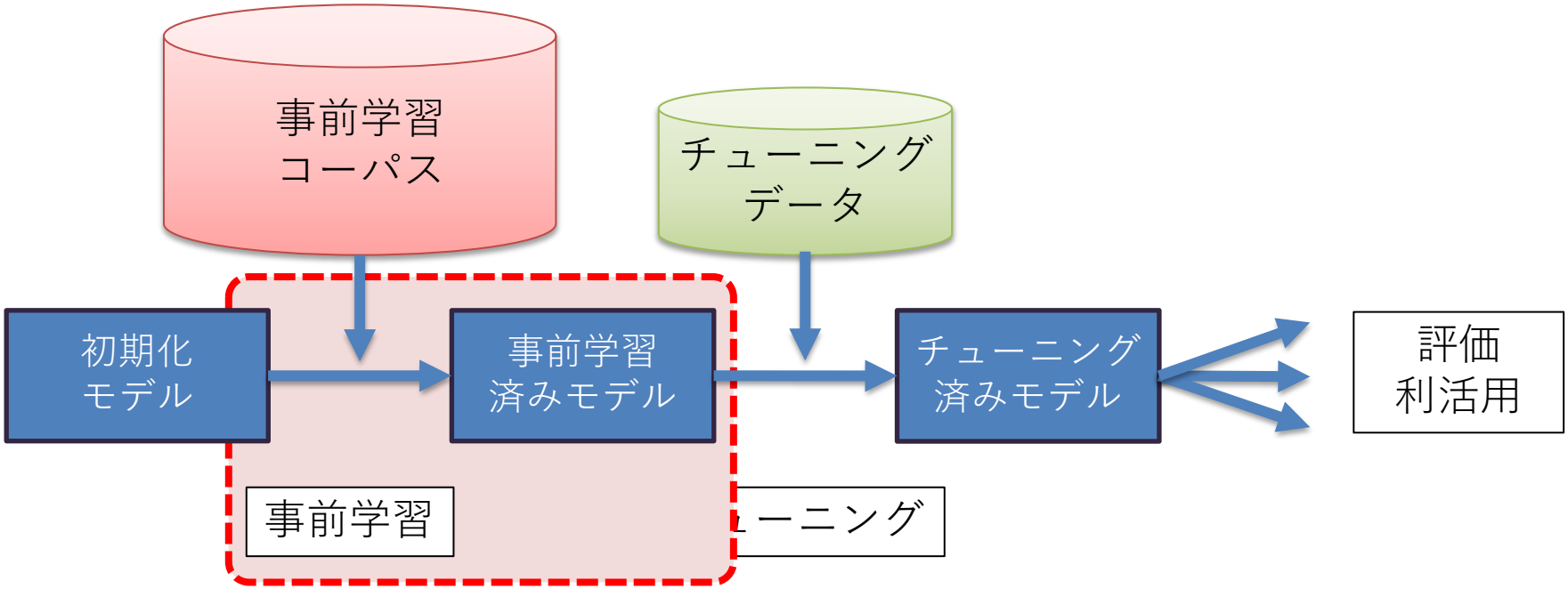
2024.09.25 (Wed)
15:35-16:35
https://www.nii.ac.jp/event/2024/0925.html

LLMシンポジウム 2024 技術講演

大規模言語モデルの事前学習

鈴木潤
国立情報学研究所 客員教授
東北大学 言語AI研究センター センター長・教授

大規模言語モデル(LLM)構築の流れ



LLM-jpでの事前学習の取り組み

- 2023-05から活動開始
  - 第一目標：13B級モデルを300Bトークンデータで学習
    - GPT-3 [2005.14165] Language Models are Few-Shot Learners

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4} ← 第一目標
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4} ← 第二目標

- 2023-10-20：LLM-jp v1をリリース
  - モデル，学習コーパス，ツールなど
  - 第二目標：175B級モデルを約1.3Tトークンデータで学習 (準備開始)
- 2024-04-15：NIIでのGENIAC期間開始
  - 172Bモデルを約2.1Tトークンデータで学習
- 2024-12頃：172Bモデル(完成版)を公開予定

LLM事前学習の状況

● 事前学習用ツール

⇨ 多くはGitHub上に公開
無料で利用可能

例

- Megatron-LM (&Megatron-Core)
  https://github.com/NVIDIA/Megatron-LM
  NVIDIA社が提供するツール　3D Parallelism, Transformer Engine, fp8 hybridなど高速な学習を実現するためのコードが揃っている
- Megatron-DeepSpeed
  https://github.com/microsoft/Megatron-DeepSpeed
  上記Megatron-LMにMicrosoft社が開発する高速かつ効率的な学習ライブラリDeepSpeedを結合
- llm-foundry https://github.com/mosaicml/llm-foundry
  元MosaicML社が提供するコード　上記ツールとは違った高速化手法が実装されている
- llm-recipes https://github.com/okoge-kaz/llm-recipes
  元MosaicML社が提供するコード　上記ツールとは違った高速化手法が実装されている
- ...

● 事前学習用コーパス

⇨ 多くはHF Datasets上にて公開
無料で利用可能

例

日本語：

- LLM-jp-corpus v1 https://github.com/llm-jp/llm-jp-corpus
  中身はWikipediaとmC4の日本語セクション　日本語 約137.5Bトークン
- LLM-jp-corpus v2 https://gitlab.llm-jp.nii.ac.jp/datasets/llm-jp-corpus-v2
  中身はWikipediaとCommon Crawlから取得　日本語 約286Bトークン
- LLM-jp-corpus v3 https://gitlab.llm-jp.nii.ac.jp/datasets/llm-jp-corpus-v3
  中身はWikipedia, Common Crawl, NDL, KAKENから取得　日本語 約579Bトークン

英語：

- Pile https://huggingface.co/datasets/EleutherAI/pile 約300Bトークン
- Dolma https://huggingface.co/datasets/allenai/dolma 約3Tトークン
- Fineweb https://huggingface.co/datasets/HuggingFaceFW/fineweb
  約13Tトークン
- RedPajama V2 https://huggingface.co/datasets/togethercomputer/RedPajama-Data-V2 約30Tトークン
- DCLM http://data.commoncrawl.org/contrib/datacomp/index.html 約240Tトークン
- ...

あとは学習を回せばいいだけ？

- LLM 事前学習の特徴と難しさ

LLM 事前学習の特徴と難しさ

- 事前学習用のデータ/ツールの取得は容易

  ⇨ ただちに実行?

  Yes: if　10Bパラメータ以下 程度のモデル

  比較的多くの試行がなされており知見も多くの論文で報告され共有されつつある

  No: if 100Bパラメータ超 のモデル

100Bパラメータ超のモデルなど

- 大規模モデルの場合
  - 小さいモデルでの知見が役に立たない場合が多々ある
  - 大きいモデルを構築できる組織 少 & 回数 少
  - サービス提供されているモデルの学習詳細は完全には公開されていない
    - 例：GPT-4, Gemini, Claude

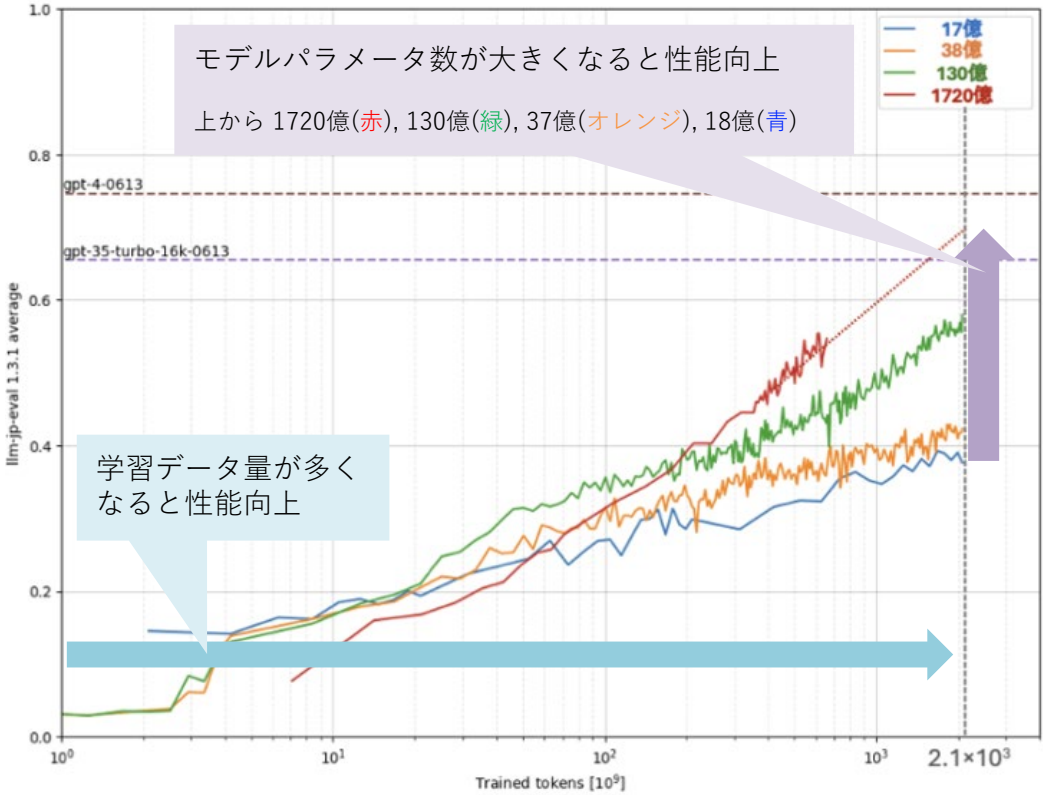
=> 直接的に使える知見がそれほど多くない

大規模な計算リソース（+膨大な予算）が必要

- 学習データ量/モデルパラメータ数 増 => 性能向上

スケール則 (Scaling Laws)

https://llmc.nii.ac.jp/topics/llm-jp-172b/



計算リソースと費用の例

BLOOM 176Bパラメータモデル(2022)

https://arxiv.org/abs/2211.05100

- 計算機：A100 (80GB) x8 を **48 nodes**
  = 合計384GPUs
- 学習データ量：366Bトークン
- 学習時間：3.5ヶ月(105日)

GCPを用いた仮定での概算コスト：$987 x 105日x 48 nodes
= **$4,974,480**

Llama3.1 405Bパラメータモデル (2024)

https://arxiv.org/abs/2211.05100

- 計算機： H100 (80GB) x8 を **2,000 nodes**(最大)
  = 合計16,000GPUs
- 学習データ量：15Tトークン
- 学習時間：3ヶ月弱 (81日)

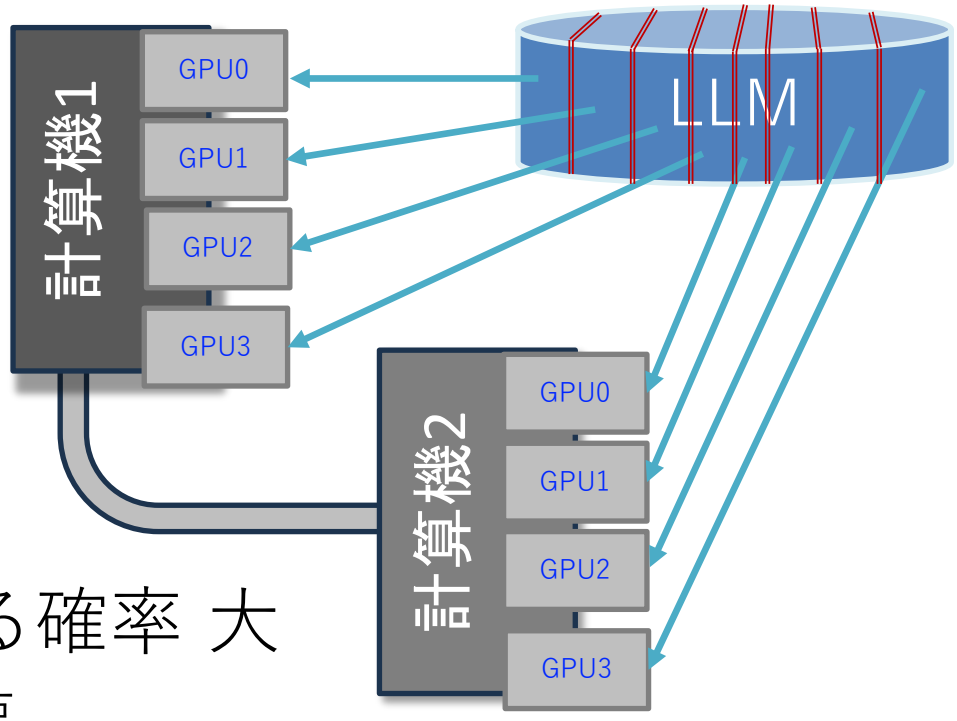
GCPを用いた仮定での概算コスト：$2,154 x 81日 x 2,000 nodes
= **$348,948,000**

(計算の効率化も含む)

計算機のHW制約への対応と障害対応

100B超級の大規模モデルの場合

- 1GPUの記憶容量(例:80GB) ＜ モデルサイズ となる
  - モデルを分割して分配
  - 高速化に適した分割/分配
  - ⇨ 利用する計算機環境とモデルに依存して決定
  - ⇨ 個別に試行錯誤して良い設定を見つける必要あり



利用する計算機環境が大きくなれば

- 計算機のトラブルに遭遇する確率 大
  - 正常に動いているか監視が必要
  - 障害の切り分けや対応が求められることも
  - 必要に応じて手動で実行スクリプトの再起動等が必要

考慮すべき学習設定が多数

- モデルサイズに依存して変更/調整が必要

例： 学習率計画の中にある最大学習率の設定

GPT-3

https://arxiv.org/pdf/2005.14165

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Llama-3.1

https://ai.meta.com/research/publications/the-llama-3-herd-of-models/

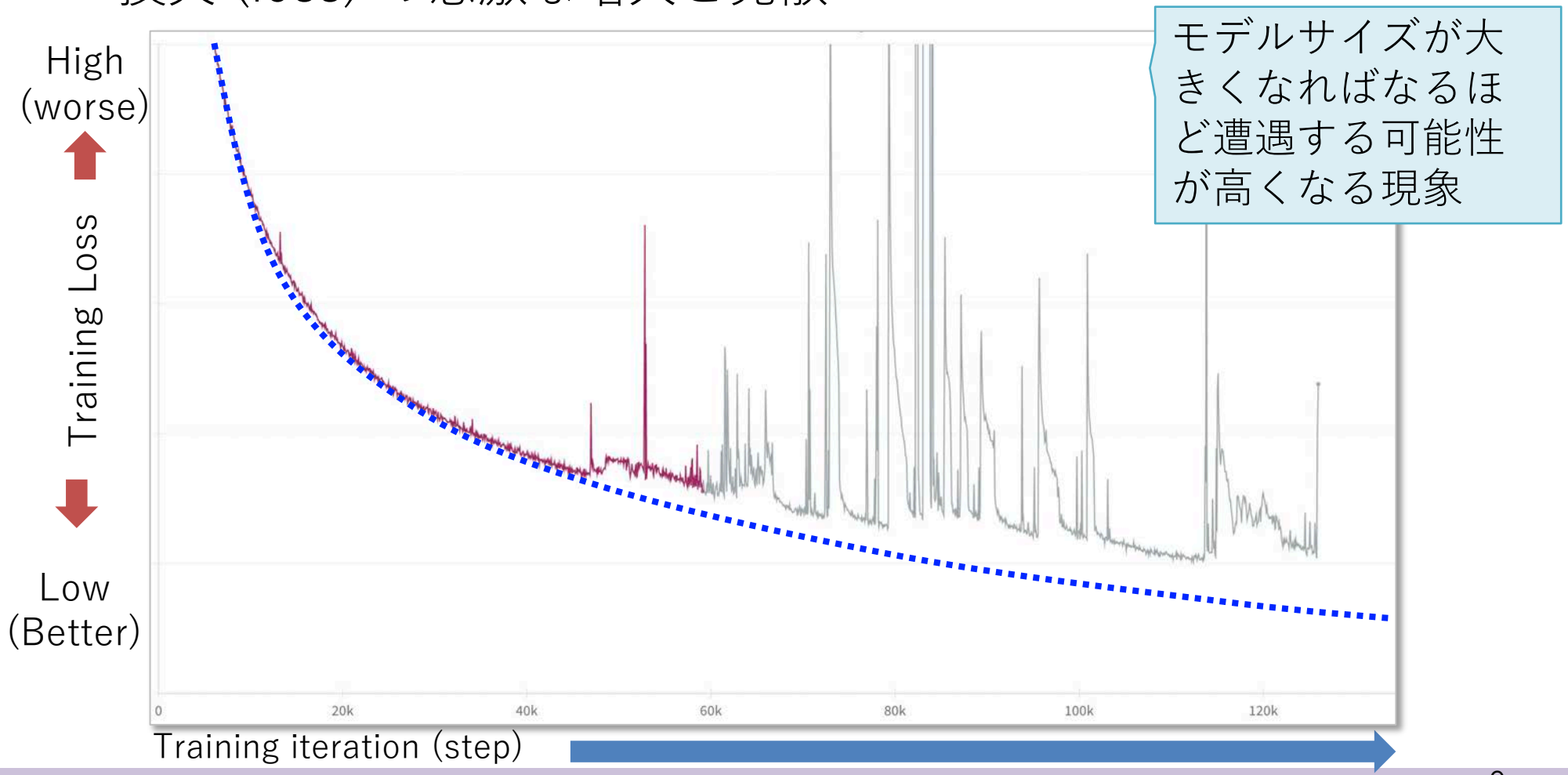
	8B	70B	405B
Layers	32	80	126
Model Dimension	4,096	8192	16,384
FFN Dimension	14,336	28,672	53,248
Attention Heads	32	64	128
Key/Value Heads	8	8	8
Peak Learning Rate	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	RoPE ($\theta = 500,000$)		

最大学習率以外にも
- Global batch size
- 学習率スケジューラー
- 多段階学習を導入するかの判断
- モデルの初期値
- 学習の安定化手法を導入するかの判断
- ・・・

といった観点を考慮する必要がある

学習の安定性問題

- 大規模モデルでは学習が進まない現象に遭遇
  - 損失 (loss) の急激な増大と発散



モデルサイズが大きくなればなるほど遭遇する可能性が高くなる現象

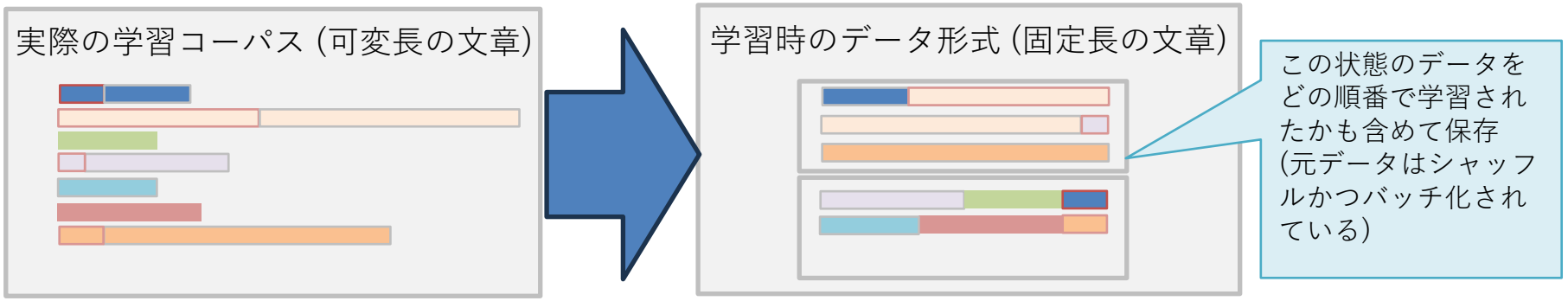
まとめ：LLMの事前学習

- LLM-jpでの取り組み
  - 172Bモデル (2.1Tトークンデータ) の構築
- LLMの事前学習の特徴と難しさ
  - 直接的に使える知見がそれほど多くない / 小さいモデルでの知見が役に立たない場合が多々ある
    - 大規模な計算リソース (＋膨大な予算) が必要
    - 計算機のハードウェア制約への対応 (計算の効率化含む) と障害対応
    - 考慮すべき学習設定が膨大
    - 損失発散の課題
- LLM-jpでは (大規模モデルの事前学習に関しても) 得られた知見を随時公開/共有していく予定

- オープンサイエンスへの取り組み
  事前学習の透明性/再現性

事前学習の透明性/再現性

- 学習時に学習コーパス（可変長データ）は固定の系列長に合わせて分割/結合される



実際学習に使われた状態をファイルに出力

=> （学習データの意味で）事前学習を
どこでも だれでも 再現可能

=> あらゆる組織で事後検証が可能

応用例：検索システム