SB Intuitions

日本語ModernBERTの構築

言語処理学会第31回年次大会 併設ワークショップ JLR2025
日本語言語資源の構築と利用性の向上
2025年3月14日

SB Intuitions株式会社

塚越 駿, 李 聖哲, 福地 成彦, 柴田 知秀

日本語ModernBERTの構築

SB Intuitions

- 大規模日英コーパスを用いた**日本語ModernBERTを構築**
  - 多様なアプリケーションに対応するため複数サイズを用意
  - Tokenizerは**Sarashina**と同じSentencePieceのみに依存
- 構築したモデルはHuggingFace上でMITライセンスで公開

HuggingFace ID	#Params	Dim.	#Layers
sbintuitions/modernbert-ja-30m	37M	256	10
sbintuitions/modernbert-ja-70m	70M	384	13
sbintuitions/modernbert-ja-130m	132M	512	19
sbintuitions/modernbert-ja-310m	315M	768	25

INDEX
目次

なぜこの時代にBERT？

SB Intuitions

- 実用上はBERTも現役
  - 埋め込みモデルやコーパスの品質推定モデルなど
  - 小回りの効く特化モデルの構築によく利用される
- 分類タスクではデコーダ系モデルよりエンコーダ系モデルが効率的
  - 多くのLLMはcausal attentionを利用するので片方向の文脈しか読めない
  - BERT等のエンコーダ系モデルは双方向の文脈を読める

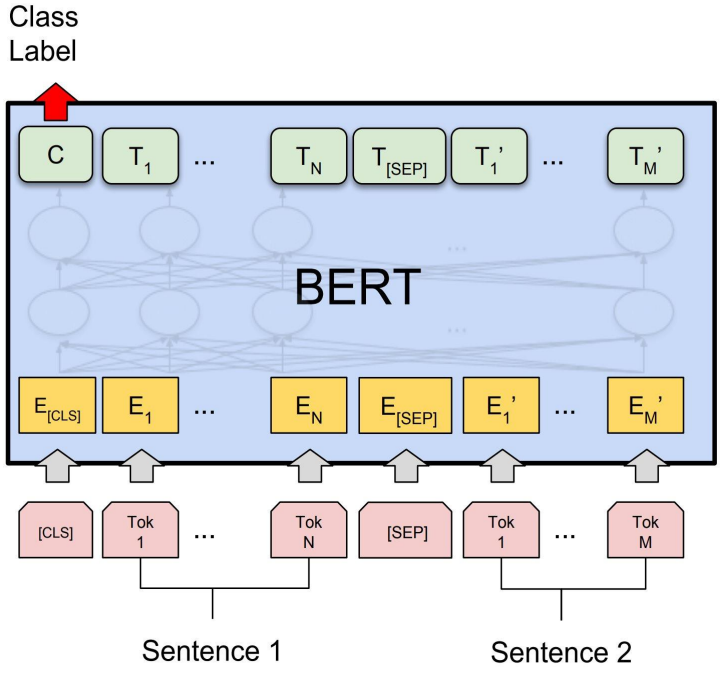
BERT

SB Intuitions

- 2018年に公開されたエンコーダモデル
- マスク穴埋めタスクで事前学習することで汎用的な言語知識を獲得
- 事前学習→Fine-tuningというパラダイムを築く

課題

- アーキテクチャが古くなりつつある
  - LLM開発に伴う進歩が取り入れられていない
- 入力系列長が512と短い
  - 埋め込みモデル等の用途には不十分



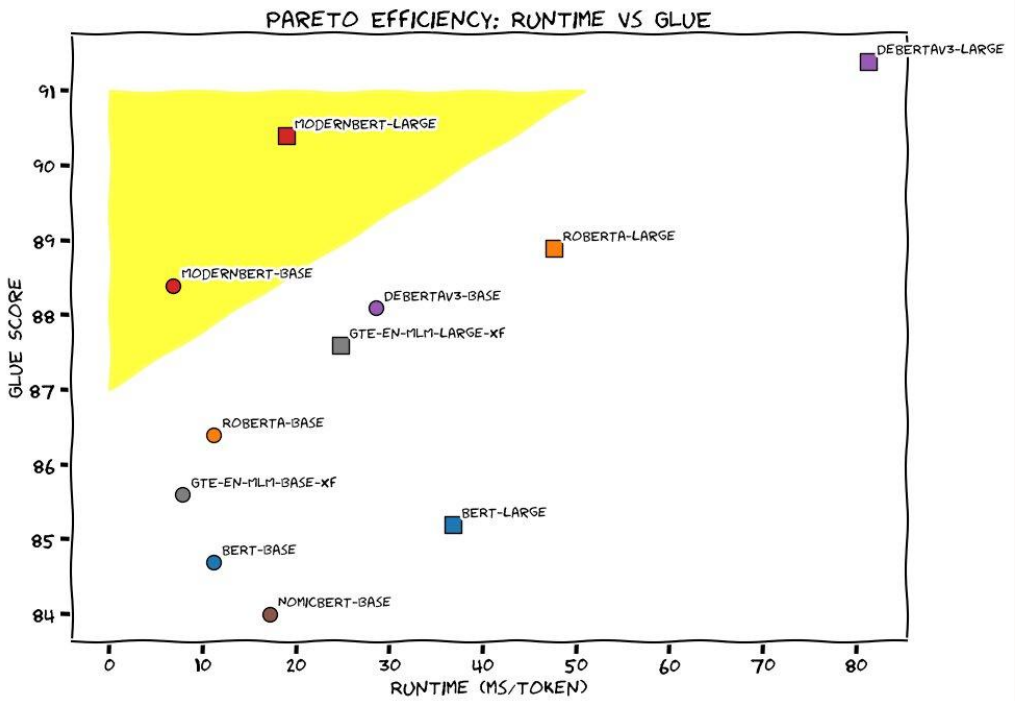
図は Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. '18 より引用

ModernBERT

SB Intuitions

- [Warnerら, ‘24]により提案
  - Llama等のLLM開発で蓄積された知見をBERTに持ち込んだ
  - 長系列も効率的に処理できるようなアーキテクチャを設計
- 実行時間・性能の双方に優れる

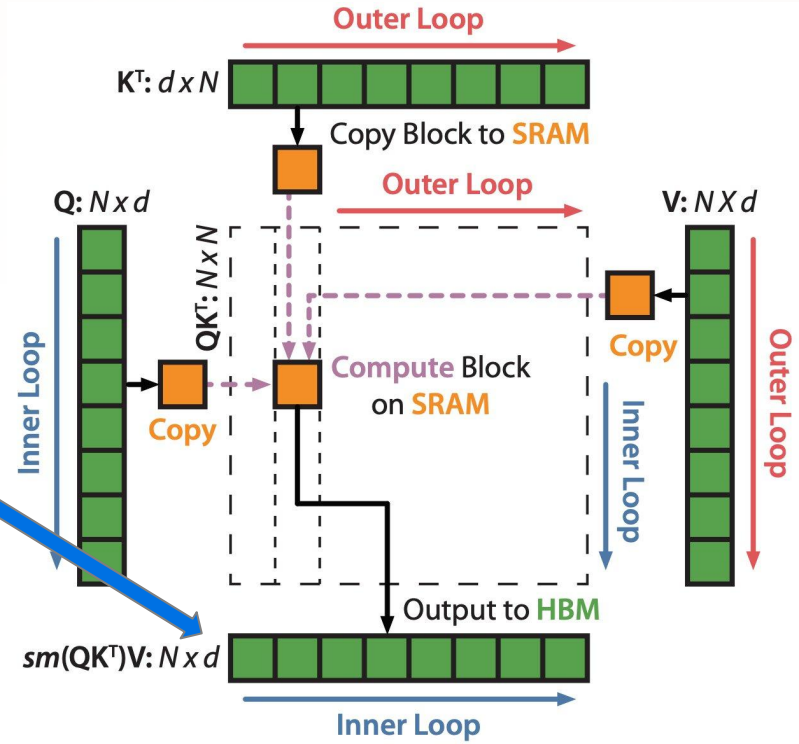
- 重要な要素技術は以下
  - **FlashAttention**
  - **RoPE (回転位置埋め込み)**
  - **Local Attention**
  - GeGLU



FlashAttention

SB Intuitions

- Attention機構の高速な実装
  - アルゴリズムレベルの改善ではなく
    ハードウェアを意識した実装レベルの改善
  - 近似はなく、厳密なAttention
- Query, Key, Valueの計算順を工夫
  - 遅いHBM経由の入出力を減らす
- PyTorchでナイーブに実装した
  Attentionと比較して3倍以上の高速化
  - 特に長系列の処理効率に優れる

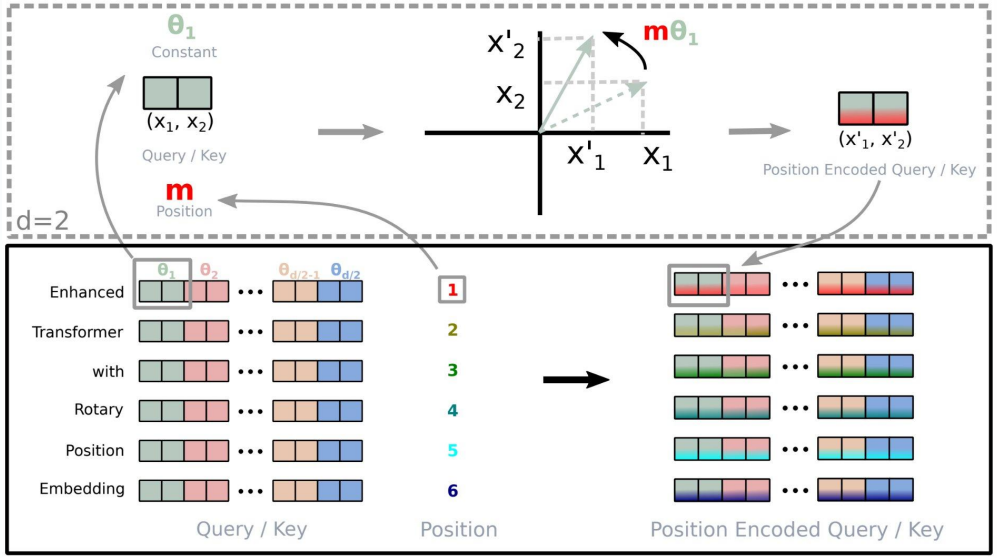


図は Dao et al., FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awarenes. '22 より引用

RoPE: 回転位置埋め込み

SB Intuitions

- 近年のLLMにおけるデファクトスタンダードの位置埋め込み手法
- トークンの埋め込み表現を回転させることで位置を表現
  - 系列中の位置によって回転の角度を変化させる
  - 従来の絶対位置埋め込みより長い系列長・良い外挿性能を実現

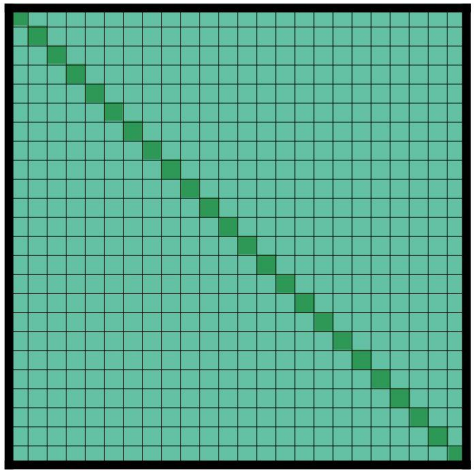


図は Su et al., RoFormer: Enhanced Transformer with Rotary Position Embedding. '21 より引用

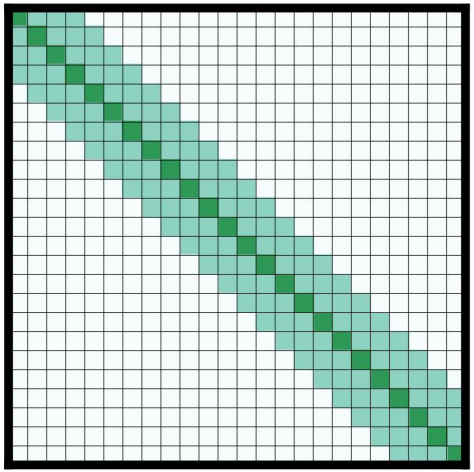
Local Attention (Sliding Window Attention)

SB Intuitions

- あるトークンの周辺トークンのみコンテキストとして考慮するAttention
  - 通常のAttentionでは系列全体をAttentionの計算対象にする
- FlashAttentionでサポートされており高速に動作



Global Attention



Local Attention

図は Beltagy et al., Longformer: The Long-Document Transformer. '20 より引用

ModernBERT: モデル構造

SB Intuitions

- Transformer Encoderを積む構成はBERTと同様
  - LLMと異なり、BERT系のAttentionは双方向
- Global Attention 1層とLocal Attention 2層を交互に
  - Global Attentionで文脈を読み、Local Attentionで周辺情報を効率的に処理する
  - 最初の層と最終層はGlobal Attention
- 層数を多く、次元数を小さくしたslim構成
  - [Tayら, '21] によればパラメータ数が同じ時の性能は **幅が狭くて深いモデル > 幅が広くて浅いモデル**
  - 層数を増やす方向にパラメータを割いた方がよい
  - ModernBERTもBERTより層数を増大
  - 代わりにMLP層のパラメータ数を削減

英語ModernBERT: 構築手順

SB Intuitions

- 3つの段階に分けてモデルを訓練
  - 全ての過程でマスク穴埋めのみ実施、次文予測 (Next Sentence Prediction) は排除
- **事前学習 (1.72T tokens)**
  - 大規模コーパスを用いて系列長1024で訓練
  - マスク率は30%
- **系列長拡張: Phase 1 (250B tokens)**
  - 系列長を1024→8192に伸ばして長い系列に対応できるよう学習
  - マスク率は30%
- **系列長拡張: Phase 2 (50B tokens)**
  - さらに高品質な長系列データをアップサンプリングして仕上げの学習
  - マスク率は**30%**

日本語ModernBERTの構築

既存の日本語BERTと課題

SB Intuitions

- 東北大BERT や Studio Ousia 日本語LUKE を始め多数のモデルが存在
  - 日本語NLPを支えてきた重要な貢献

課題

- **入力系列長が短い**
  - 多くのモデルで入力系列長が512
  - 8192程度の長系列が入力可能なモデルが望ましい
- **モデルサイズのバリエーションが少ない**
  - 多くの既存モデルはbase, largeサイズのみ
  - 実用上は30M〜300M程度の様々なサイズのモデルを使い分けたい

日本語ModernBERTの構築 (再掲)

SB Intuitions

- 大規模日英コーパスを用いた**日本語ModernBERTを構築**
  - 多様なアプリケーションに対応するため複数サイズを用意
  - Tokenizerは**Sarashina**と同じSentencePieceのみに依存
- 構築したモデルはHuggingFace上でMITライセンスで公開

HuggingFace ID	#Params	Dim.	#Layers
sbintuitions/modernbert-ja-30m	37M	256	10
sbintuitions/modernbert-ja-70m	70M	384	13
sbintuitions/modernbert-ja-130m	132M	512	19
sbintuitions/modernbert-ja-310m	315M	768	25

日本語ModernBERTの構築: 手順

SB Intuitions

- オリジナルのModernBERTに倣い、3つの段階に分けてモデルを訓練
  - 実験設定は多少変更しつつ実験的に改善を目指す

- **事前学習 (日英3.5T tokens)**
  - マスク率: 30%
- **系列長拡張: Phase 1 (高品質日英400B tokens)**
  - Best-fit packing利用
  - マスク率: 30%
- **系列長拡張: Phase 2 (高品質日本語150B tokens)**
  - Sequence packingは不使用
  - マスク率: **15%** (評価実験の節で解説)

日本語ModernBERTの構築: 学習環境

SB Intuitions

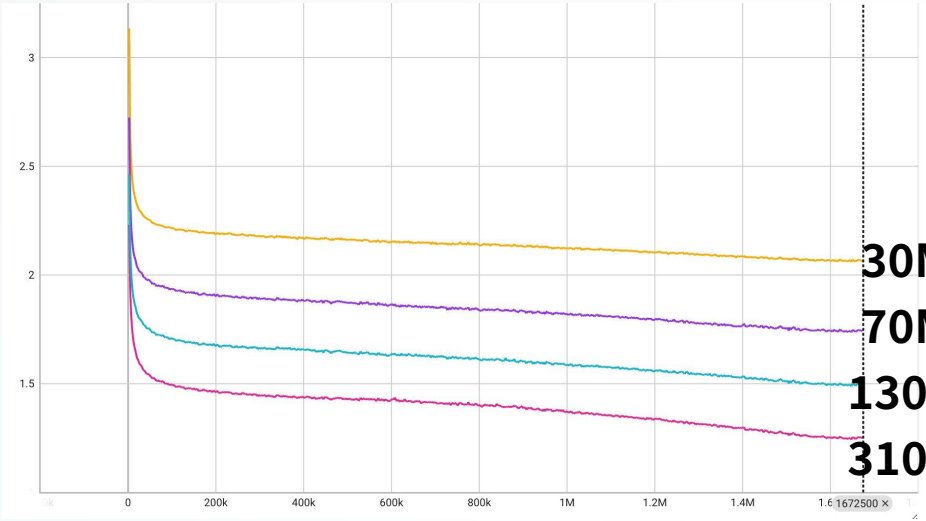
- シンプルなマルチノード学習を実施
- 主に下記ライブラリ/技術を使用
  - HuggingFace Transformers
  - HuggingFace Accelerate
  - DeepSpeed ZeRO 2

- 事前学習 (3.5T tokens) に用いたノード・時間
  - **130M: A100 x 128枚 で 約120時間**
  - **310M: H100 x 256枚 で 約70時間**

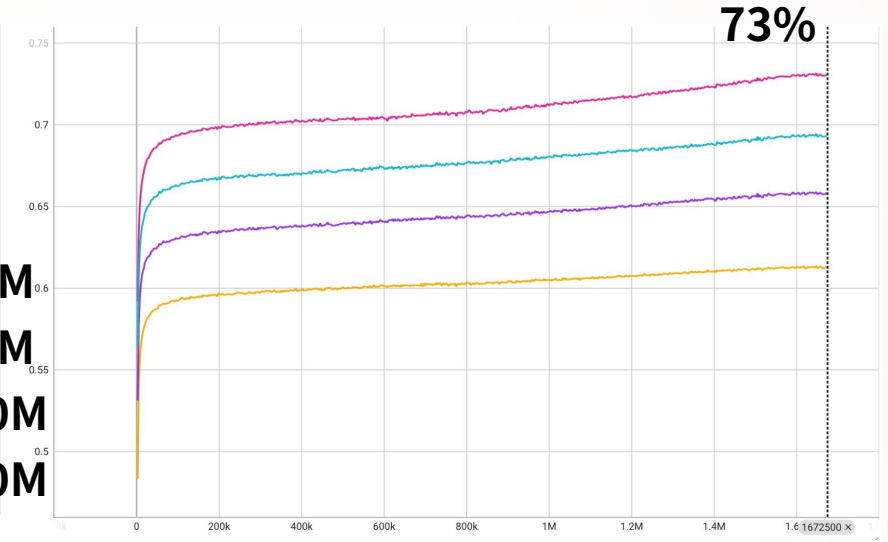
日本語ModernBERTの構築: 訓練過程

SB Intuitions

- 事前学習時のマスク穴埋め (マスク率30%) の損失・正解率をプロット
  - 性能は安定して向上し続ける
  - Sudden Drops [Chenら, '24] は観測せず



開発セット損失



開発セット正解率

評価実験

性能評価: 評価タスク

SB Intuitions

- 日本語の分類・回帰タスク12種類を選定して網羅的に評価
  - 評価結果は `2025.03.07` 時点のもの

- 知識系タスク: **JCommonsenseQA**、**RCQA**
- 日本語の統語的評価: **JCoLA**
- 自然言語推論タスク: **JNLI, JSICK, JSNLI, 京大RTE**
- 意味的類似度タスク: **JSTS**
- 文分類タスク: **Livedoorニュース**, **LLM-jp Toxicity**, **WRIME v2**, **MARC-ja**

⚠ tokenizerの違いにより、NERなどトークン分類系タスクでの評価は未実施
  - ModernBERT-Jaは形態素解析器を利用しないtokenizerのため

評価結果: smallサイズのモデル

SB Intuitions

- 既存の小規模モデルと比較して高い性能を発揮
  - 特に日本語知識系タスク・自然言語推論タスクで高い性能

Model	#Params	JComQA	JCoLA	JNLI	12タスク平均
ModernBERT-Ja-30M	37M	80.95	78.85	88.69	**85.67**
ModernBERT-Ja-70M	70M	85.65	80.26	90.33	**86.77**
mMiniLMv2-L6-H384	107M	60.34	78.61	86.24	81.53
mMiniLMv2-L12-H384	118M	62.70	78.61	87.69	82.59
LINE/DistillBERT	68M	76.39	81.04	87.49	85.32

評価結果: base・largeサイズのモデル

SB Intuitions

- 既存のモデルと比較して高い性能
  - 特に310Mモデルは既存モデルと比較して最高性能（2025.03.07時点）

Model	#Params	JComQA	JCoLA	JNLI	12タスク平均
ModernBERT-Ja-130M	132M	91.01	84.18	92.03	88.95
ModernBERT-Ja-310M	315M	93.53	84.81	92.93	89.83
東北大BERT-base v3	111M	82.82	81.50	89.68	86.74
東北大BERT-large v2	337M	86.93	82.89	92.05	88.36
日本語LUKE-large	414M	88.01	84.34	92.37	88.94

分析: 学習量

SB Intuitions

- 130Mモデルについて構築段階ごとの性能を評価

- 段階を経るごとに性能が向上
- 要因
  - 段階ごとにコーパスが異なること
  - Sequence Packing戦略の違い

段階	JComQA	JCoLA	JNLI	12タスク平均
事前学習	86.83	78.69	91.09	87.25
系列長拡張 Phase 1	87.38	80.10	90.90	87.50
系列長拡張 Phase 2	91.01	84.18	92.03	88.95

分析: マスク率

SB Intuitions

- マスク率30%の場合...
  - 入力系列長が100なら30トークンが**<mask>**になっている
  - あまりにも**<mask>**が多いと日本語の自然さが損なわれるのでは？
- 系列長拡張 Phase 2で性能比較
  - マスク率を30%、15%と変えて実験
- 全般的に性能向上が見られた
  - 特に知識系タスク・日本語の自然さを評価するタスクで性能向上

マスク率	JComQA	JCoLA	JNLI	12タスク平均
30%	89.58	82.31	92.39	88.54
15%	90.21	83.58	91.54	88.70

分析: エポック数

SB Intuitions

- 系列長拡張 Phase 2はオリジナルのModernBERTでは50B tokensのみ
  - 我々のモデルでは開発損失が改善し続ける様子が確認できた
- 学習量を増やすことで性能が向上するのでは？
  - Epoch数を1→３に増加させてモデルを構築
- 全般的な性能向上が見られた
  - 特に知識系タスク・日本語の自然さを評価するタスクで性能向上

マスク率	JComQA	JCoLA	JNLI	12タスク平均
30% 1 epoch	89.58	82.31	92.39	88.54
15% 1 epoch	90.21	83.58	91.54	88.70
15 % 3 epoch	**91.01**	**84.18**	**92.03**	**88.95**

性能評価: パラメータ数と性能の関係

SB Intuitions

- ModernBERT-Jaのパラメータ数ごとの性能を比較
  - 開発中の1.4Bも含めて評価
- エンコーダモデルも安心してスケーリングして良さそう
  - 特に知識系タスクはモデルサイズと綺麗に相関

モデルサイズ	JComQA	JCoLA	JNLI	12タスク平均
30M	80.95	78.85	88.69	85.67
70M	85.65	80.26	90.33	86.77
130M	91.01	84.18	92.03	88.95
310M	93.53	84.81	92.93	89.83
1.4B (開発中)	95.64	86.33	93.07	91.14

性能評価: SLMとの比較

SB Intuitions

- 小規模なLLM (SLM) を分類モデル用にfine-tuningして性能を比較
  - 文末に文末トークンを付加、その位置の埋め込み表現を分類器に入力
  - LoRAなどの手法は利用せずフルパラメータでfine-tuning
- 130Mでも1Bクラスのモデルに匹敵、310Mは1Bクラスのモデルを上回る
  - **5~10倍のパラメータ効率**

モデルサイズ	12タスク平均
ModernBERT-Ja-130M	88.95
ModernBERT-Ja-310M	**89.83**
Qwen/Qwen2-1.5B-Instruct	87.68
pfnet/plamo-2-1b	87.37
sbintuitions/sarashina2.1-1b	89.03
llm-jp/llm-jp-3-1.8b-instruct3	89.41

性能評価: Sarashina 2.2との比較

SB Intuitions

- 先週公開した0.5B、1B、3BのSLM
- 先ほどと同様、分類モデル用にfine-tuningして性能を比較
  - 文末に文末トークンを付加、その位置の埋め込み表現を分類器に入力
- **ModernBERT-Ja 1.4Bモデルは Sarashina 2.2 3Bモデルも上回る**性能

モデルサイズ	12タスク平均
ModernBERT-Ja-130M	88.95
ModernBERT-Ja-310M	89.83
ModernBERT-Ja-1.4B (開発中)	**91.14**
Sarashina2.2-0.5b-instruct-v0.1	88.33
Sarashina2.2-1b-instruct-v0.1	89.08
Sarashina2.2-3b-instruct-v0.1	**90.83**

日本語ModernBERT: 使用上の注意

SB Intuitions

- 評価ベンチマークで最良になった回数が最多の学習率を算出
- 今回構築したModernBERT-Jaは既存BERTより小さめの学習率が良さそう
  - 利用する際は既存日本語BERTより小さい学習率でのFine-tuningを推奨

モデル	最多学習率
ModernBERT-Ja-30M	2e-05
ModernBERT-Ja-70M	1e-05
ModernBERT-Ja-130M	1e-05
ModernBERT-Ja-310M	1e-05
東北大BERT-base v3	3e-05
東北大BERT-large v2	1e-05
Studio Ousia 日本語LUKE-large	1e-05

まとめ

SB Intuitions

- 長系列に対応できる日本語に特化した**ModernBERT-Ja**シリーズを構築
  - 30M, 70M, 130M, 310Mと異なるモデルサイズを提供
- BERT系モデルの網羅的な日本語評価を実施
  - 構築したModernBERT-Jaはそれぞれのモデルサイズで最良の性能
  - **特に310Mは既存モデルと比較しても最高性能**（2025.03.07時点）

今後の展望

- 日本語における長系列での評価
  - 評価ベンチマークの整備から
- 大規模モデルからの知識蒸留
- **MeCab**や**Juman++**と組み合わせた際の性能評価
- その他の詳細はテックブログをお待ちください！

直感を、知性へ


SB Intuitions