



O'REILLY®
オライリー・ジャパン

機械学習エンジニアのための Transformers

最先端の自然言語処理ライブラリ
によるモデル開発



Lewis Tunstall
Leandro von Werra 著
Thomas Wolf
中山 光樹 訳

輪講ルール

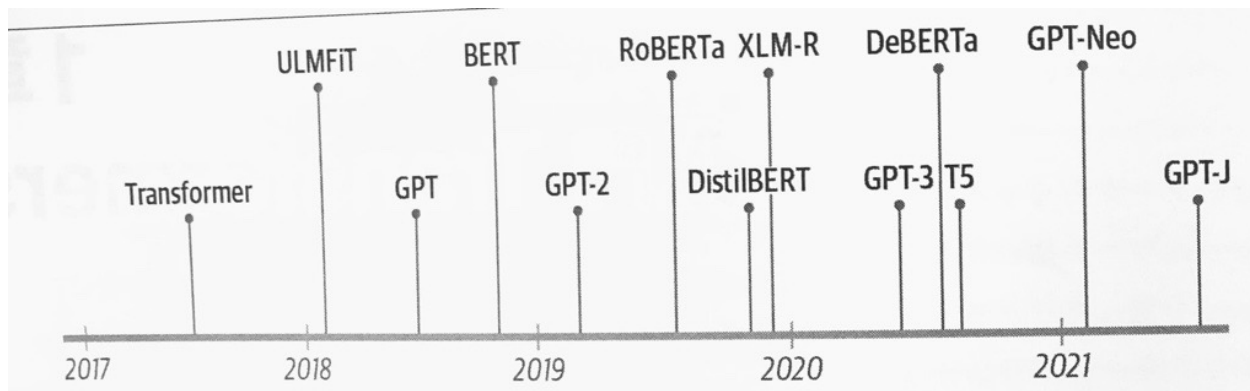
- 担当することを目的化しない
 - 自分で理解して人に説明できるようになることが目的
 - 書籍だけでわからない用語や概念があればググって調べることも必要
 - 担当範囲をあらかじめ決めない
 - 1週間で理解できた範囲をのみをスライド化（無理にページ数を稼いだせいで質が落ちることを避ける）
- 使用されている図はできる限り載せる

日付：2023/6/28, 担当者：田口

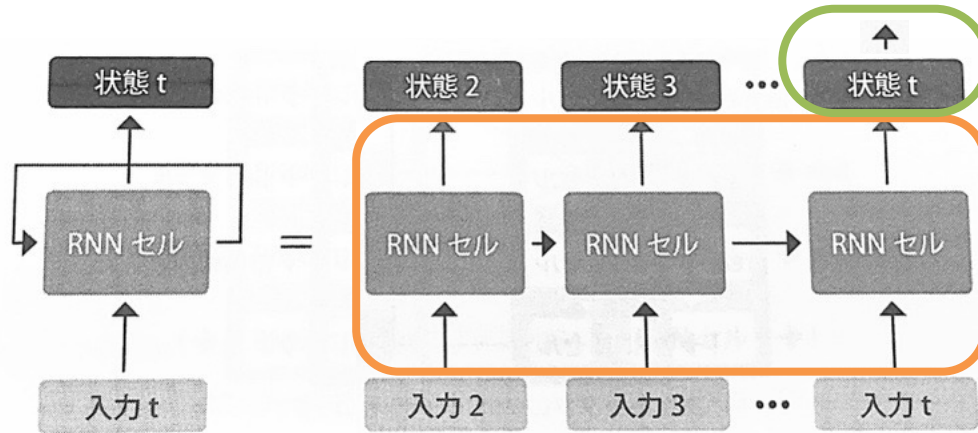
- 担当箇所：p.1~p.15
 - 1章 入門Transformers
 - 1.1 エンコーダ・デコーダのフレームワーク
 - 1.2 アテンション機構
 - 1.3 NLPにおける転移学習
 - 1.4 Hugging Face Transform
 - 1.5 Transformerの応用を巡る旅

~~~~~

## ✓ Transformerの年表

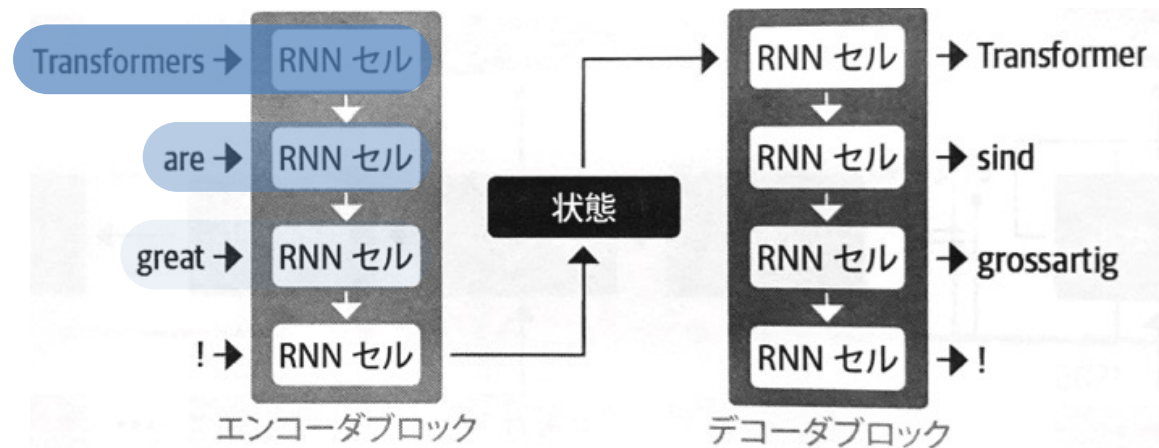


# 1.1 エンコーダ・デコーダのフレームワーク



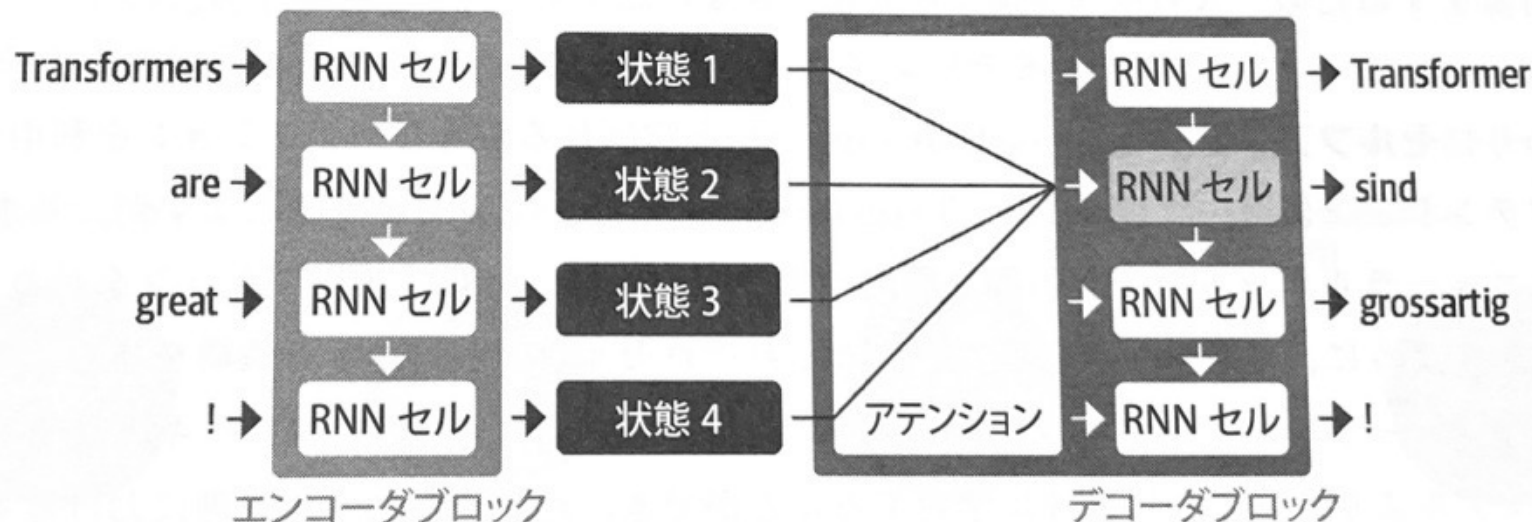
RNNの入力・出力アーキテクチャ

- エンコーダ・デコーダ  
(**encoder-decoder**)
- Seq2Seq  
(sequence-to-sequence)



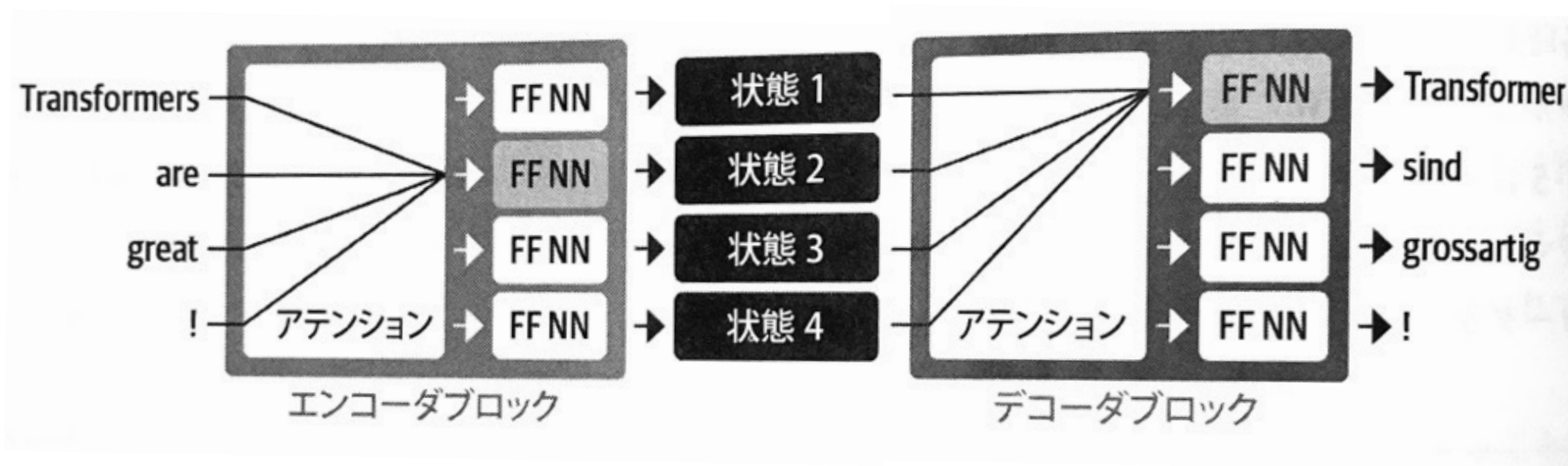
⚠ シンプルだが最後の隠れ状態で最初の方の情報が失われる

## 1.2 アテンション機構(1/2)



- RNNのためのアテンション
    - デコーダがエンコーダのすべての隠れ状態にアクセスできる
- (⚠①)入力系列が並列処理できない  
⇒デコーダがエンコーダの各状態にデコードする時刻ごとに異なる  
「重み」=「**アテンション**」を割り当てて優先順位付けする

## 1.2 アテンション機構(2/2)



- RNNのためのアテンション機構

- エンコーダ・デコーダにリカレントモデルを使用する

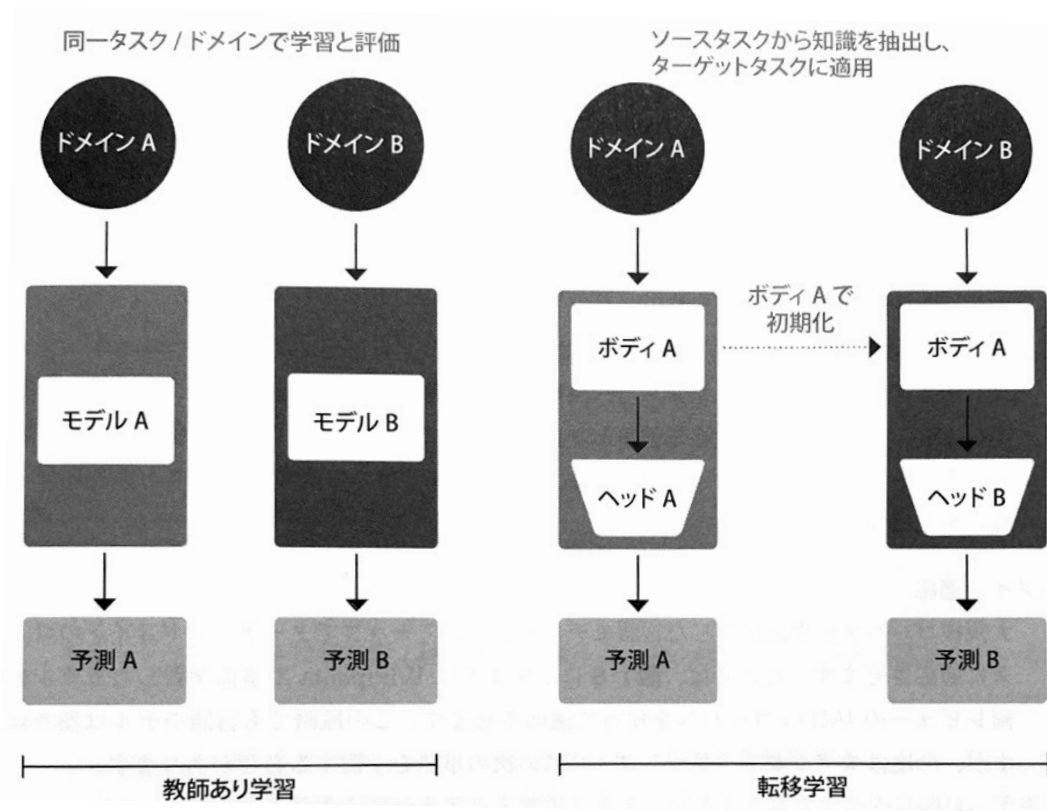
(⚠②)逐次に計算するため入力系列全体で並列化できない

⇒ **セルフアテンション**を利用

# 1.3 NLPにおける転移学習(1/2)

- 転移学習：あるタスクを学習させたニューラルネットワークを，新しいタスクに適応させる

= ファインチューニング

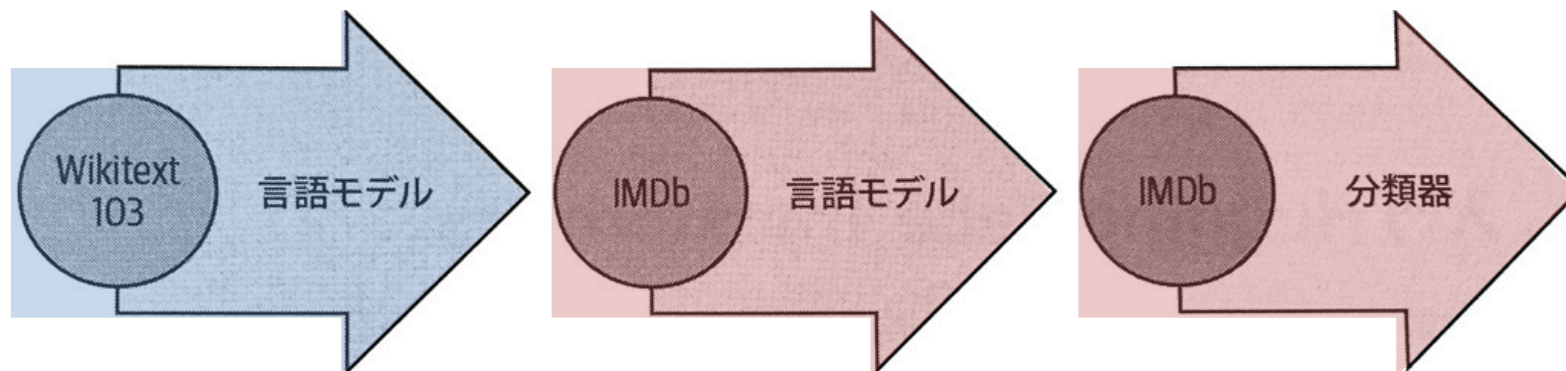


ボディ: 学習データドメインの  
広範な特徴を学習した  
ネットワーク

ヘッド: タスクに特化した  
ネットワーク

# 1.3 NLPにおける転移学習(2/2)

- ULMFiT : 事前学習と転移学習によるモデル



- Transformer : セルフアテンションと転移学習
  - GPT : Transformerのデコーダ部分のみを用いて, ULMFiTと同じアプローチで学習
  - BERT : Transformerのエンコーダ部分とマスク言語モデルで学習

# 1.4 Hugging Face Transformers

- Hugging Face Transformers

Transformerを開発するためのライブラリ  
学習済みモデルやデータセットのリポジトリ

1. PytorchやTensorFlowでモデルを実装
2. サーバから事前学習済みの重みを読み込む
3. 入力を前処理して得られた出力をタスクに応じた後処理
4. データローダを実装しモデルを学習するための損失関数と最適化アルゴリズムを定義する

# 1.5 Transformerの応用を巡る旅

Hugging Face TransformersのAPIを用いてできること

- テキスト分類：感情の2値分類(センチメント分析)
- 固有表現認識：実世界のオブジェクトをテキストから抽出
- 質問応答：与えられた質問をテキストから抽出する
- 要約：関連するすべての事実を含んだテキスト生成
- 翻訳
- テキスト生成：テキストの自動補完

# 日付：2023/?/?, 担当者：

- 担当箇所：p.??~p.??
  - ?章 Transformers
    - ?.1 エンコーダ・デコーダのフレームワーク
    - ?.2 アテンション機構
    - aa
    - aa
    - aa



# 1.1 エンコーダ・デコーダのフレームワーク